Logistic Regression

Ulrich Halekoh, Jørgen Vinslov Hansen, Søren Højsgaard
Biometry Research Unit
Danish Institute of Agricultural Sciences

March 31, 2006

Contents

1 Analysis of the budworm moth data

Reading the data and defining a variable `logdose` (called d in the lecture).

```
library(dataRep)
data(budworm)
budworm$logdose <- log(budworm$dose)
```

The response for a logistic model is preferably defined as a two column matrix. The first column contains the observed cases. The second column contains the number of non-cases:

```
budworm$y <- cbind(budworm$ndead, budworm$ntotal - budworm$ndead)
```

Alternatively one can use the `with` function to tell R where to find the data.

```
budworm$y <- with(budworm, cbind(ndead, ntotal - ndead))
```

If we had Bernoulli data, i.e. the cases were either 0 or 1, a response 0-1 vector of these cases would be sufficient.

2 Estimates and confidence intervals for the parameters

The model the logits of the binomially distributed number of killed moths is

$$M0 : y_{S,d} \sim bin(n_{Sd}, \pi_{S,d}) \quad S = F(emale), M(ale)$$

as

$$\text{logit}(\pi_{S,d}) = \mu + \alpha_S + \gamma d \tag{1}$$

where $d = \log(D)$

The estimation is performed with the `glm` function.

We assume that the data are binomially distributed. This assumption is specified via the `family` argument. With this family it is assumed by default, that we model the logits of the probabilities. This means, that the `link`-function is the logit.

The fit is obtained by

```
M0 <- glm(y ~ 1 + sex + logdose, data = budworm, family = binomial(link = logit))
```

Note, we need not to specify the logit-link, because it is assumed by default. Likewise we need not to specify the intercept by `1` because it is included by default. If we do not define y beforehand we can write model fit also as

```
M0 <- glm(cbind(ndead, ntotal - ndead) ~ 1 + sex + logdose,
      data = budworm, family = binomial(link = logit))
```

We print the parameter estimate table

```
coef(summary(M0))
```

```
            Estimate Std. Error z value   Pr(>|z|)
(Intercept)  -3.4732    0.46852 -7.4130 1.2344e-13
sexmale       1.1007    0.35583  3.0935 1.9782e-03
logdose       1.5353    0.18910  8.1190 4.7015e-16
```

The likelihood-ratio confidence intervals for the parameters are obtained by (you must load the CRAN package MASS)

```
library(MASS)
confint(M0)
```

We produce a smarter table adding the estimates

```
tab1 <- coef(summary(M0))
tab2 <- confint(M0)
tab.est <- cbind(tab1[, "Estimate"], tab2)
colnames(tab.est) <- c("Estimate", colnames(tab2))
```

```
            Estimate    2.5 %  97.5 %
(Intercept)  -3.4732 -4.45824 -2.6137
sexmale       1.1007  0.41924  1.8205
logdose       1.5353  1.18720  1.9319
```

3 Interpretation of parameters

3.1 Continuous covariate

The estimate for γ is $\hat{\gamma} = 1.54$. Increasing the log-dose by 1 will therefore increase the odds-ratio of dying by a factor of $\exp(\hat{\gamma})$:

```
exp(M0$coef["logdose"])
```

Question: How much is the odds-ratio increased if you

1. increase the log-dose by going from log-dose d to log-dose $d + 3$?
2. double the log-dose ?

3.2 Factor regressor

The regressor `sex` is represented in (1) by the two parameters α_{female} and α_{male}. Using the reference (or treatment) coding, one parameter is set to 0. R choose automatically the level of the factor that is lexically ordered first. In the present case $alpha_{female} = 0$ You can actively change another reference level by using the `relevel` function. i.e.

```
relevel(budworm$sex, "male")
```

The model is fitted by

```
M0 <- glm(y ~ 1 + sex + logdose, family = binomial, data = budworm)
```

The `1` represents the μ in the model and is included by default.

The parameter estimates with the setting $\alpha_{female} = 0$ are

```
summary(M0)$coef[, "Estimate"]
```

```
(Intercept)     sexmale     logdose
    -3.4732      1.1007      1.5353
```

Their interpretation is given in table 1.

Table 1: Interpretation of Parameters

R output	parameters	interpretation
`(Intercept)`	$\mu = \mu + \alpha_{female}$	intercept for females, i.e. $\text{logit}(\pi_{female,d=0})$
`(Intercept)+sexmale`	$\mu + \alpha_{male}$	intercept for males, i.e. $\text{logit}(\pi_{male,d=0})$
`sexmale`	$\alpha_{male} - \alpha_{female} = \alpha_{male}$	logit difference or log odds-ratio of mortality of male to female
`logdose`	γ	log odds-ratio of mortality of increasing $\log(D)$ by 1 (or multiplying D by $\exp(1) = 2.7$.)

4 Estimation or prediction of a probability

4.1 Prediction at all the covariate values in the data set

The prediction at all the covariate values in the data set from which the model has been fitted is easily obtained as

```
pred.prob <- predict(M0, type = "response")
```

The `type=response` specification requests the prediction of probabilities. The default is `type=link` which gives predictions for the linear predictor η,

```
pred.logits <- predict(M0, type = "link")
```

which are in this case the logits and applying the `plogis` function $\frac{\exp(\eta)}{1+\exp(\eta)}$ to the log-odds

```
plogis(pred.logits)
```

we obtain again the probabilities.

4.2 Prediction at specific values of the covariates

If we want to know the probability at the dose $D = 27$ for a female moth we create the data

```
new <- data.frame(sex = "female", logdose = log(27))
```

and get the prediction

```
pred.prob <- predict(M0, newdata = new, type = "response")
```

To obtain a confidence interval for this probability it is better first to obtain the Wald-confidence interval on the logit scale. We predict using the additional argument `se` to get the standard error.

```
pred.eta <- predict(M0, newdata = new, type = "link", se = TRUE)
```

The confidence interval for $\eta_{female,d=1.9}$ is then

```
CI.lower <- pred.eta$fit - 1.96 * pred.eta$se.fit
CI.upper <- pred.eta$fit + 1.96 * pred.eta$se.fit
```

or shorter

```
CI <- pred.eta$fit + 1.96 * c(-1, 1) * pred.eta$se.fit
```

The confidence interval for $\eta_{female,d=1.9}$ is obtained by applying the logistic function in R by `plogis`:

```
plogis(CI)
```

```
[1] 0.71932 0.90318
```

An alternative way to compute the prediction and the confidence interval is via the `esticon` function of the CRAN-package `doBy`. Based on the parameter estimates of the model

```
M0$coef
```

```
(Intercept)     sexmale     logdose
    -3.4732      1.1007      1.5353
```

the log-odds for females at dose=27 is written as

$$1 \cdot \mu + 1 \cdot \alpha_{female} + \gamma \log(27)$$

The vector $(1, 1, \log(27))$ is the coefficient vector to be used in the `esticon` function.

```
library(doBy)
pred.logit <- esticon(M0, cm = c(1, 1, log(27)))
```

The probability and a 95% confidence interval is then given by

```
plogis(as.numeric(pred.logit[c("Estimate", "Lower.CI",
      "Upper.CI")]))
```

```
[1] 0.93630 0.86986 0.96999
```

4.3 Plotting raw and predicted probabilities

We want to plot the proportions and the estimated probabilities in one plot to get an insight into the quality of the model fit. The proportions are

```
budworm$prop <- budworm$ndead/budworm$ntotal
```

We want to predict the models at a dense grid of the log dose to be able to draw a smooth line. We create a data frame where for each sex we have 50 dose observations between log dose 0 and log(32):

```
logdose = log(seq(1, 32, l = 50))
new <- expand.grid(sex = c("male", "female"), logdose = logdose)
```

The function `expand.grid` crosses all the values of the first variable with all the values of the second.

The predicted probabilities are obtained as

```
pred.prob <- predict(M0, newdata = new, type = "response")
```

First we set up the plot by using the `type='n'` option:

```
plot(prop ~ dose, data = budworm, xlab = "dose", ylab = "probability",
      type = "n", col = 1, pch = 16)
```

Then we add the points for the proportions. We use here the formula based version of the plot function because it allows a sub-setting. Because of an error in the function, it is advantageous to specify the data set from which the plotted data are taken

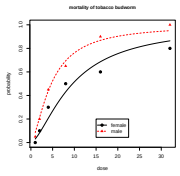
```
points(prop ~ dose, data = budworm, subset = c(sex == "female"),
      col = 1, pch = 16)
points(prop ~ dose, data = budworm, subset = c(sex == "male"),
      col = 2, pch = 17)
```

Now we add the predicted probabilities

```
dat <- data.frame(pred.prob = pred.prob, new, dose = exp(new$logdose))
lines(pred.prob ~ dose, data = dat, subset = c(sex == "female"),
      col = 1, pch = 16)
lines(pred.prob ~ dose, data = dat, subset = c(sex == "male"),
      col = 2, pch = 17)
```

Finally, add a legend and a title

```
legend(15, 0.2, legend = c("female", "male"), col = c(1,
      2), pch = c(16, 17), lty = c(1, 2))
title("mortality of tobacco budworm", cex.main = 0.9)
```



4.4 Model selection

We calculate the likelihood ratio test between model

$$M0 : \text{logit}(\pi_{S,d}) = \mu + \alpha_S + \gamma d$$

and

$$M1 : \text{logit}(\pi_{S,d}) = \mu + \alpha_S + \gamma_S \gamma d$$

First fitting the models

```
M0 <- glm(cbind(ndead, ntotal - ndead) ~ 1 + sex + logdose,
      data = budworm, family = binomial)
M1 <- glm(cbind(ndead, ntotal - ndead) ~ 1 + sex + logdose +
      sex:logdose, data = budworm, family = binomial)
```

Then comparing them by the `anova` function

```
anova(M0, M1, test = "Chisq")
```

Equivalently we can use the `drop1` function on th lager model M1

```
drop1(M1, test = "Chisq")
```

```
tab <- anova(M0, M1, test = "Chisq")
library(xtable)
print(xtable(tab, digits = c(0, 0, 2, 0, 2, 3), caption = "Likelihood ratio test comparing model M
      label = "tab:logist-liktest-budworm"), caption.placement = "top",
      file = "table/logist-liktest-budworm.tex")
```

4.5 Goodness-of-fit

The residual deviance of model M0 is given by

```
M0$deviance
```

```
[1] 6.757
```

and the Pearson X^2 is the sum of the Pearson residuals

```
res <- residuals(M0, type = "pearson")
X2 <- sum(res^2)
```

```
[1] 5.306
```

The residual degrees of freedom are

```
M0$df.residual
```

```
[1] 9
```

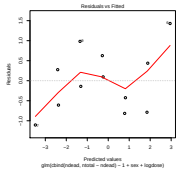
A p-value for the goodness-of.fit test that the model M0 is appropriate for the data is given by

```
1 - pchisq(X2, M0$df.residual)
```

4.6 Model checking

For model checking a plot of the deviance residuals against the predicted logits is obtained by

```
plot(M0, which = 1)
```



If you want to differentiate in the plot between the sexes use

```
plot(M0, which = 1, pch = c(16, 1)[budworm$sex])
```