Lecture 5: Comparing logistic regression models and effect modification

Interaction

Evaluation of possible interaction, also referred to as effect modification, is an important part of any regression analysis. We will look back at the passive smoking example here.

```
smoking.data <- data.frame(y=c(161,117,120,111),
                    n=c(291,241,200,266),
                    passive=c(1,0,1,0),
                    smoker=c(1,1,0,0))
# intercept only
mod0 <- glm(cbind(y,n-y) ~ 1,
            data=smoking.data, family=binomial)

# just passive exposure to smoking
mod1 <- glm(cbind(y,n-y) ~ passive,
            data=smoking.data, family=binomial)

# just (active) smoking
mod2 <- glm(cbind(y,n-y) ~ smoker,
            data=smoking.data, family=binomial)

# active and passive smoking, no interaction
mod3 <- glm(cbind(y,n-y) ~ passive+smoker,
            data=smoking.data, family=binomial)

# active and passive smoking, plus interaction
mod4 <- glm(cbind(y,n-y) ~ passive*smoker,
            data=smoking.data, family=binomial)
```

We can carry out likelihood ratio tests of various hypotheses based on these nested models.

```
# analysis of deviance (likelihood ratio testing)

# test whether passive smoking is associated with cancer
anova(mod0,mod1,test='LRT') # "test" option gives the likelihood ratio test
comparing these nested models

## Analysis of Deviance Table
##
## Model 1: cbind(y, n - y) ~ 1
## Model 2: cbind(y, n - y) ~ passive
##   Resid. Df Resid. Dev Df Deviance  Pr(>Chi)
## 1         3    18.4772
## 2         2     3.4362  1   15.041 0.0001052 ***
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# test whether model 1 can be improved on by adding active smoking and the
interaction
anova(mod1,mod4,test='LRT')
```

```
## Analysis of Deviance Table
##
## Model 1: cbind(y, n - y) ~ passive
## Model 2: cbind(y, n - y) ~ passive * smoker
##   Resid. Df Resid. Dev Df Deviance Pr(>Chi)
## 1         2     3.4362
## 2         0     0.0000  2   3.4362   0.1794
```

```
# test for the interaction
anova(mod3,mod4,test='LRT')
```

```
## Analysis of Deviance Table
##
## Model 1: cbind(y, n - y) ~ passive + smoker
## Model 2: cbind(y, n - y) ~ passive * smoker
##   Resid. Df Resid. Dev Df Deviance Pr(>Chi)
## 1         1     3.2788
## 2         0     0.0000  1   3.2788  0.07018 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# to compare non-nested models, use AIC or BIC
AIC(mod1)
```

```
## [1] 31.21009
```

```
AIC(mod2)
```

```
## [1] 45.53443
```

```
BIC(mod1)
```

```
## [1] 29.98268
```

```
BIC(mod2)
```

```
## [1] 44.30702
```

Compare the estimated models with and without interaction terms.

```
# without interaction
summary(mod3)
```

```
##
## Call:
## glm(formula = cbind(y, n - y) ~ passive + smoker, family = binomial,
```

```
##     data = smoking.data)
##
## Deviance Residuals:
##       1        2        3        4
## -0.8315   0.9069   1.0052  -0.8687
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -0.22618    0.10833  -2.088  0.03681 *
## passive      0.48720    0.12849   3.792  0.00015 ***
## smoker       0.05108    0.12873   0.397  0.69150
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 18.4772  on 3  degrees of freedom
## Residual deviance:  3.2788  on 1  degrees of freedom
## AIC: 33.053
##
## Number of Fisher Scoring iterations: 3
```

```
# with interaction
summary(mod4)
```

```
##
## Call:
## glm(formula = cbind(y, n - y) ~ passive * smoker, family = binomial,
##     data = smoking.data)
##
## Deviance Residuals:
## [1]  0  0  0  0
##
## Coefficients:
##                Estimate Std. Error z value Pr(>|z|)
## (Intercept)     -0.3339     0.1243  -2.685 0.007246 **
## passive          0.7394     0.1905   3.881 0.000104 ***
## smoker           0.2758     0.1791   1.540 0.123569
## passive:smoker  -0.4674     0.2585  -1.808 0.070570 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance:  1.8477e+01  on 3  degrees of freedom
## Residual deviance: -4.1744e-14  on 0  degrees of freedom
## AIC: 31.774
##
## Number of Fisher Scoring iterations: 2
```

Contrasts

To determine the odds ratios in an interaction logistic model, we would like to set up contrasts, or functions of linear combinations of the parameter estimates. To do this, we need to specify a contrast matrix, which is essentially a matrix of 0's and 1's, with the 1's in (column) positions corresponding to elements of the coefficient vector that we want to add up. Each row of this matrix corresponds to a different contrast.

```
# look at the coefficients
coef(mod4)
```

```
##    (Intercept)        passive         smoker passive:smoker
##     -0.3338949      0.7393600      0.2757873     -0.4673825
```

```
# set up the contrast matrix
L <- rbind(c(0,1,0,0), # OR of passive for nonsmokers
           c(0,1,0,1), # OR of passive for smokers
           c(0,0,1,0), # OR of smoker for no exposure to passive smoke
           c(0,0,1,1)  # OR of smoker for exposure to passive smoke
           )

# log odds ratios for each contrast
lor.cont <- L %*% coef(mod4)
# get standard errors for contrasts (on log scale)
se.lor.cont <- sqrt(diag(L %*% vcov(mod4) %*% t(L)))

# contrast odds ratios
exp(lor.cont)
```

```
##          [,1]
## [1,] 2.094595
## [2,] 1.312558
## [3,] 1.317568
## [4,] 0.825641
```

```
# lower confidence limits on odds scale
exp(lor.cont-qnorm(1-.05/2)*se.lor.cont)
```

```
##           [,1]
## [1,] 1.4419089
## [2,] 0.9320212
## [3,] 0.9275424
## [4,] 0.5729885
```

```
# upper confidence limits on odds scale
exp(lor.cont+qnorm(1-.05/2)*se.lor.cont)
```

```
##          [,1]
## [1,] 3.042721
## [2,] 1.848464
```

```
## [3,] 1.871596
## [4,] 1.189698
```

We can use the delta method to get standard errors on odds scale. The derivative of exp(x) is just exp(x), so we multiply the odds ratio by the standard error on the log odds scale.

```
exp(lor.cont)*se.lor.cont
```

```
##           [,1]
## [1,] 0.3990406
## [2,] 0.2292847
## [3,] 0.2359592
## [4,] 0.1538815
```

Remember that using this estimated standard error will give you different confidence intervals than exponentiating the confidence interval constructed on the log odds scale.

This method can also be used to get predicted probabilities and confidence intervals. We just have to modify the L (contrast) matrix.

```
# set up the contrast matrix (this is different now)
L <- rbind(c(1,0,0,0), # nonsmokers with no passive exposure
           c(1,1,0,1), # nonsmokers with passive exposure
           c(1,0,1,1), # smokers with no passive exposure
           c(1,1,1,1)  # smokers with passive exposure
           )

# logit-scale predicted probabilities for each contrast
linpred.cont <- L %*% coef(mod4)
# get standard errors for contrasts (on logit scale)
se.linpred.cont <- sqrt(diag(L %*% vcov(mod4) %*% t(L)))

# contrast predicted probabilities
plogis(linpred.cont)
```

```
##           [,1]
## [1,] 0.4172932
## [2,] 0.4845256
## [3,] 0.3715694
## [4,] 0.5532646
```

```
# lower confidence limits on probability scale
plogis(linpred.cont-qnorm(1-.05/2)*se.linpred.cont)
```

```
##           [,1]
## [1,] 0.3594854
## [2,] 0.3817395
## [3,] 0.2759550
## [4,] 0.4956913
```

```
# upper confidence limits on probability scale
plogis(linpred.cont+qnorm(1-.05/2)*se.linpred.cont)
```

```
##           [,1]
## [1,] 0.4774674
## [2,] 0.5886377
## [3,] 0.4784223
## [4,] 0.6094439
```

Standard errors are a bit trickier since the derivative of the `plogis()` function isn't as simple as `exp()`. However, it isn't too complicated:

```
plogis(linpred.cont)*plogis(-linpred.cont)*se.linpred.cont
```

```
##            [,1]
## [1,] 0.03023465
## [2,] 0.05355348
## [3,] 0.05231642
## [4,] 0.02914373
```

Profile likelihood confidence intervals

Profile likelihood confidence intervals are analogous to likelihood ratio confidence intervals in multiparameter models. They involve holding the parameter of interest fixed at a range of values, and at each value in this range, maximizing the model log-likelihood with respect to the remaining parameters. The maximized log-likelihoods then define a curve as a function of the parameter of interest, which can be used to find what values of the parameter would not be rejected by a likelihood ratio test.

This is simple to calculate in R.

```
# profile likelihood confidence intervals
confint(mod4)
```

```
## Waiting for profiling to be done...
```

```
##                       2.5 %      97.5 %
## (Intercept)    -0.57954646 -0.09152811
## passive         0.36790767  1.11528275
## smoker         -0.07481989  0.62765538
## passive:smoker -0.97511511  0.03847117
```

```
# Wald confidence intervals
confint.default(mod4)
```

```
##                       2.5 %      97.5 %
## (Intercept)    -0.57759830 -0.09019153
## passive         0.36596784  1.11275221
## smoker         -0.07521675  0.62679131
## passive:smoker -0.97398280  0.03921785
```