Categorical Data Analysis
Lecture 10

Graduate School of Advanced Science and Engineering
Rei Monden

2025.1.9

Building and applying logistic regression models

Building and applying logistic regression models

Today we will focus on

- model selection

and

- model fit

for logistic regression models.

Model selection

Model selection

Choosing an adequate model may be hard when there are many predictors available.

We must balance two opposing ideals:

- Fit: The model should be complex enough to fit the data well.
- Parsimony: The model should be simple enough to interpret and generalize beyond our sample.

Model selection

There is also a difference between confirmatory and exploratory analyses:

- Confirmatory: When theory dictates which effects we wish to test for inclusion. We can then compare two models – with and without the effects of interest.
- Exploratory: When there is no strong theory, we may have potentially many models available to compare.

Model selection

Q: How many predictors p can we include in a logistic regression model?

A: This actually depends on the proportion of 0s and 1s in the response variable y.

In general, use this simple rule-of-thumb as guidance:

> *The data set should contain at least 10 outcomes of each type for every explanatory variable.*

Example:
Suppose our data set has $n = 200$ observations such that y has 24 0s and 176 1s.

In this case, the rule-of-thumb above suggests including no more than 2 predictors.

Then we have at least $2 \times 10 = 20$ outcomes of each type (0 or 1).

Model selection

Like ordinary regression, be careful about including predictors that are highly correlated (i.e., multicollinearity).

This makes effects 'fight against each other'.
Parameter estimates and their associated SEs will be poorly estimated and the model will be quite unstable as a consequence.

Model selection – Example in R

width	weight	color	spine	y
28.3	3.05	2	3	1
22.5	1.55	3	3	0
26.0	2.30	1	1	1
⋮	⋮	⋮	⋮	⋮
28.0	2.625	1	1	0
27.0	2.625	4	3	0
24.5	2.000	2	2	0

173 rows in total.

Predictors:

- *width*, shell width in cm (continuous variable).
- *weight*, the crab's weight in kg
- *color*, shell color (categorical: 1 = medium light, 2 = medium, 3 = medium dark, 4 = dark). The darker, the older.
- *spine*, spine condition
  (categorical: 1 = both good, 2 = one broken, 3 = both broken)

Outcome: *y*, binary (0 = female has no satellites; 1 = has satellites).

Model selection – Example in R

```r
# Import data frame from file:
crab.df <- read.table("Crabs.dat", header = TRUE)

# Model 1: Main effects - width and color
crab.fit1 <- glm(y ~ width + factor(color),
                 family = binomial(link = "logit"),
                 data   = crab.df
                 )
summary(crab.fit1)
logLik(crab.fit1)
deviance(crab.fit1)

# Model 2: Main effects - width, weight, color, and spine
crab.fit2 <- glm(y ~ width + weight + factor(color) + factor(spine),
                 family = binomial(link = "logit"),
                 data   = crab.df
                 )
summary(crab.fit2)
logLik(crab.fit2)
deviance(crab.fit2)
```

Model selection – Example in R

Model 1:

```
Coefficients:
                Estimate Std. Error z value Pr(>|z|)
(Intercept)    -11.38519    2.87346  -3.962 7.43e-05 ***
width            0.46796    0.10554   4.434 9.26e-06 ***  (!!!)
factor(color)2   0.07242    0.73989   0.098    0.922
factor(color)3  -0.22380    0.77708  -0.288    0.773
factor(color)4  -1.32992    0.85252  -1.560    0.119
```

Model 2:

```
Coefficients:
               Estimate Std. Error z value Pr(>|z|)
(Intercept)    -8.06501    3.92855  -2.053   0.0401 *
width           0.26313    0.19530   1.347   0.1779       (!!!)
weight          0.82578    0.70383   1.173   0.2407
factor(color)2 -0.10290    0.78259  -0.131   0.8954
factor(color)3 -0.48886    0.85312  -0.573   0.5666
factor(color)4 -1.60867    0.93553  -1.720   0.0855 .
factor(spine)2 -0.09598    0.70337  -0.136   0.8915
factor(spine)3  0.40029    0.50270   0.796   0.4259
```

Model selection – Example in R

Model 1, the simpler, fits significantly better than the null model ($\chi^2(4) = 38.30$, $p < .001$):

```
crab.fit0 <- glm(y ~ 1,
                 family = binomial(link = "logit"),
                 data   = crab.df
                 )
anova(crab.fit0, crab.fit1, test = "LRT")
```

Output:

```
-----------------------------------------------
  Resid. Df Resid. Dev Df Deviance Pr(>Chi)
2       168     187.46  4   38.301 9.71e-08 ***
-----------------------------------------------
```

Model selection – Example in R

Model 2 also fits significantly better than the null model ($\chi^2(7) = 40.56$, $p < .001$):

```
anova(crab.fit0, crab.fit2, test = "LRT")
```

Output:

```
---------------------------------------------
  Resid. Df Resid. Dev Df Deviance Pr(>Chi)
2       165     185.20  7   40.557 9.848e-07 ***
---------------------------------------------
```

Model selection – Example in R

Model	Log-lik	Deviance	df
1	$L_0 = -93.73$	$D_0 = 187.46$	5
2	$L_1 = -92.60$	$D_1 = 185.20$	8

How about the comparison between models 1 and 2?

```
anova(crab.fit1, crab.fit2, test = "LRT")
```

Output:

```
----------------------------------------------
  Resid. Df Resid. Dev Df Deviance Pr(>Chi)
2       165     185.20  3    2.255   0.5212
----------------------------------------------
```

$\chi^2(3) = 2.26$, $p = .52$:
We conclude that Model 2 does not fit significantly better than Model 1.
We therefore choose Model 1.

Model selection – Example in R

Another aspect against Model 2 is that *width* and *weight* correlate highly:

```
cor(crab.df$width, crab.df$weight)
```

Output:

```
[1] 0.8868715
```

These predictors compete against each other in Model 2, which is not OK (in terms of model fit and interpretation).

Model selection – Example in R

Model 1:

```
Coefficients:
             Estimate Std. Error z value Pr(>|z|)
-----------------------------------------------------
width         0.46796    0.10554   4.434 9.26e-06 ***
-----------------------------------------------------
```

Model 2:

```
Coefficients:
             Estimate Std. Error z value Pr(>|z|)
-----------------------------------------------------
width         0.26313    0.19530   1.347   0.1779
weight        0.82578    0.70383   1.173   0.2407
-----------------------------------------------------
```

Stepwise variable selection algorithms

Stepwise variable selection algorithms are automatized algorithms to select the 'best' subset of predictors from a larger set.

There are several types:

- Forward selection: Start from the *null* model and sequentially add effects as while as it improves model fit.
- Backward elimination: Start from the *full* model and sequentially remove effects as while as it does not overly hurt model fit.
- Stepwise selection: Mixes both forward selection and backward elimination.

Stepwise variable selection algorithms

General rules:

- *Categorical predictors:*
  Recall that a categorical predictor with k groups requires $(k-1)$ code variables.
  To include (exclude) a categorical predictor, add (remove) all k code variables.
- *Only test the higher-order terms for each variable:*
  For example, a model including an interaction $x_i \times x_j$ should also include the main effects x_i and x_j.

Stepwise variable selection algorithms

Remember that...

- Different stepwise variable selection algorithms may lead to different final models.
- Such models may be difficult to interpret theoretically.
- Small sample variation may lead to different models selection.
- Effects that are *theoretically relevant* should be included, regardless of their statistical significance.
- Especially for large sample sizes, also consider *practical* instead of only *statistical* significance.

AIC and the bias/variance tradeoff

AIC and the bias/variance tradeoff

There is hardly a situation where there is the 'correct' model.

As Box famously said,

> *"All models are wrong, but some are useful."*

The secret is to choose the *best* (not necessarily correct!) model *for the purpose at hand* (e.g., prediction, explanation, etc.).

AIC and the bias/variance tradeoff

Model selection hinges on a tradeoff between bias and variance.

Consider, for example, two competing models A and B such that A is simpler than B.
Each model is preferable in *some way*:

- Model A:
  - ✓ Less *variance* of the estimated parameters across samples (due to having less parameters).
  - ✗ More *bias* in the estimated parameters (i.e., difference between true and estimated parameters).
- Model B:
  - ✓ Less *bias*.
  - ✗ More variance.

AIC and the bias/variance tradeoff

It is typically impossible to choose a model that simultaneously...

accurately captures the underlying trends and relationships (i.e., small bias)

and

generalizes well to new data (i.e., small variance).

AIC and the bias/variance tradeoff

In other words, one must balance between...

> *Model overfitting:*
> *Using complex models that capture too much noise in the observed data but that generalize poorly to new data (low bias, high variance)*

and

> *Model underfitting:*
> *Using simple models that fail to capture the signal in the observed data but that generalize well to new data (high bias, low variance).*

AIC and the bias/variance tradeoff

The bias/variance tradeoff, in particular, implies that significance tests are not enough to do good model selection.

The AIC (Akaike information criterion) is often used:

$$\mathrm{AIC} = -2(\text{log likelihood}) + 2(\text{number of parameters in model})$$

AIC and the bias/variance tradeoff

$$\text{AIC} = \underbrace{-2(\text{log likelihood})}_{(1)} + \underbrace{2(\text{number of parameters in model})}_{(2)}$$

The smaller the AIC, the better.
A suitable model is therefore one that:

- Fits the data well (i.e., high log likelihood $\Rightarrow$ (1) small).
  This implies low bias.

and

- Is relatively simple (i.e., few parameters $\Rightarrow$ (2) small).
  This implies low variance.

AIC addresses the bias/variance tradeoff!!

AIC and the bias/variance tradeoff – Example in R

```
# Model 1: Main effects - width and color
crab.fit1 <- glm(y ~ width + factor(color),
                 family = binomial(link = "logit"),
                 data   = crab.df
                 )
logLik(crab.fit1)
```

Output:

```
'log Lik.' -93.72852 (df=5)
```

$$\begin{aligned}\text{AIC} &= -2(\text{log likelihood}) + 2(\text{number of parameters in model}) \\ &= -2(-93.72852) + 2(5) \\ &= 197.457.\end{aligned}$$

AIC and the bias/variance tradeoff – Example in R

```
# Model 1: Main effects - width and color
crab.fit1 <- glm(y ~ width + factor(color),
                 family = binomial(link = "logit"),
                 data   = crab.df
                 )
AIC(crab.fit1)
```

Output:

```
[1] 197.457
```

AIC and the bias/variance tradeoff – Example in R

Let's compare Models 1 and 2:

```
# Model 1: Main effects - width and color
crab.fit1 <- glm(y ~ width + factor(color),
                 family = binomial(link = "logit"),
                 data   = crab.df
                 )
# Model 2: Main effects - width, weight, color, and spine
crab.fit2 <- glm(y ~ width + weight + factor(color) + factor(spine),
                 family = binomial(link = "logit"),
                 data   = crab.df
                 )
AIC(crab.fit1, crab.fit2)
```

Output:

```
          df     AIC
crab.fit1  5 197.457
crab.fit2  8 201.202
```

Conclusion:
Model 1 is the best in the AIC sense (best bias/variance tradeoff).

AIC and the bias/variance tradeoff

The AIC can also be used in a stepwise variable selection algorithm.

For example, in backward elimination:

1. Start with the full model including all possible predictors.
2. At each step, remove the variable that leads to the largest AIC reduction.
3. Stop once removing any other variable would lead to *increasing* the AIC.

AIC and the bias/variance tradeoff – Example in R

```r
# Model 2: Main effects - width, weight, color, and spine
crab.fit2 <- glm(y ~ width + weight + factor(color) + factor(spine),
                 family = binomial(link = "logit"),
                 data   = crab.df
                 )
library(MASS)
stepAIC(crab.fit2)
```

Output:

```
-----------------------------------------------------
Start:  AIC=201.2
y ~ width + weight + factor(color) + factor(spine)

Step:  AIC=198.21
y ~ width + weight + factor(color)

Step:  AIC=197.46
y ~ width + factor(color)
-----------------------------------------------------
```

Conclusion:
The selected model includes the effects *width* and *color*.

AIC and the bias/variance tradeoff

Alternatively, the AIC can be used on models based on *all* possible combinations of predictors.

For example, for the running example including 4 possible predictors (*width*, *weight*, *color*, *spine*), that implies computing the AIC for $2^4 = 16$ different models.

The 'best' model is the one with the smallest AIC.

AIC and the bias/variance tradeoff – Example in R

```
crab.data <- crab.df[, c("weight", "width", "color", "spine", "y")]

library(bestglm)
bestglm(crab.data, family = binomial, IC = "AIC")
```

Output:

```
-----------------------------------------------------------
Best Model:
               Estimate Std. Error   z value     Pr(>|z|)
(Intercept) -10.0708390  2.8068339 -3.587971 3.332611e-04
width         0.4583097  0.1040181  4.406056 1.052696e-05
color        -0.5090467  0.2236817 -2.275763 2.286018e-02
-----------------------------------------------------------
```

Conclusion:
The selected model includes the effects *width* and *color*.

Model checking

Model checking

One thing is to choose the 'best' model among a set of candidate models.
Other is to say that the chosen model fits the data *well*.

For example, given the three following models
(here, 'crappy' = fits the data poorly),

- Model 1: Unbelievably crappy model
- Model 2: Very crappy model
- Model 3: Crappy model,

it is entirely likely that the AIC points at Model 3.

Although Model 3 is the best of the three models, it is still *crappy*...

Thus, after model selection, it is always crucial to check model fit!!

Goodness of fit: Model comparison using the deviance

We used this idea before:

> *Compare deviances (via likelihood-ratio tests!) to compare a model ($\mathcal{H}_0$) to a more complex competitor ($\mathcal{H}_1$).*

- Rejecting $\mathcal{H}_0$ implies that we should keep the complex model, as it fits significantly better than the simpler model.
- Failing to reject $\mathcal{H}_0$ implies that we should keep the simple model, as we do not have enough evidence against it.

This is based on a significance test.
As discussed before, also consider practical significance or theory to make a final decision.

Goodness of fit: Model comparison using the deviance

```r
# Model 1: Main effects - width and color:
crab.fit1 <- glm(y ~ width + factor(color),
                 family = binomial(link = "logit"),
                 data   = crab.df
                 )

# Model 3: Further add the interaction effect:
crab.fit3 <- glm(y ~ width * factor(color),
                 family = binomial(link = "logit"),
                 data   = crab.df
                 )

deviance(crab.fit1)
deviance(crab.fit3)
anova(crab.fit1, crab.fit3, test = "LRT")
```

Goodness of fit: Model comparison using the deviance

Output:

```
[1] 187.457
[1] 183.0806
------------------------------------------------------------
  Resid. Df Resid. Dev Df Deviance Pr(>Chi)
2       165     183.08  3   4.3764   0.2236
```

$$\begin{aligned}\chi^2 &= D_1 - D_3 \\ &= 187.457 - 183.0806 \\ &= 4.38\end{aligned}$$

Conclusion:
We keep Model 1, since adding the interaction did not overly improve model fit.

Goodness of fit: Model comparison using the deviance

Another strategy is to compare observed with predicted values:

If the model fits well, then observed and predicted values should be close on average.

Here, the idea is similar to comparing 'our' model to the saturated model.
The saturated model:

- Includes as many parameters as data points.
- Fits the data perfectly (i.e., observed = predicted).

We can do this also via the likelihood-ratio test.

Goodness of fit: Model comparison using the deviance

```
# Model 1: Main effects - width and color:
crab.fit1 <- glm(y ~ width + factor(color),
                 family = binomial(link = "logit"),
                 data   = crab.df
                 )

# Model 4: Saturated model:
crab.fit4 <- glm(y ~ diag(173), # one parameter per observation
                 family = binomial(link = "logit"),
                 data   = crab.df
                 )

deviance(crab.fit1)
deviance(crab.fit4)
anova(crab.fit1, crab.fit4, test = "LRT")
```

Goodness of fit: Model comparison using the deviance

Output:

```
[1] 187.457
[1] 1.003674e-09  # 0, essentially
-------------------------------------------------------------
  Resid. Df Resid. Dev Df Deviance Pr(>Chi)
2         0       0.00 168   187.46   0.1448
```

$$\begin{aligned}\chi^2 &= D_1 - D_4 \\ &= 187.457 - 0 \\ &= 187.457\end{aligned}$$

Conclusion:
We keep Model 1: There is not enough evidence that Model 1 fits worse than the saturated model.

Goodness of fit: Model comparison using the deviance

Other idea is to generalize the chi-squared tests that we learned in Chapter 2 in the context of contingency tables.

This only works when all predictors are categorical.

Recall the *likelihood-ratio chi-squared statistic*:

$$G^2 = 2 \sum_{i,j} \text{observed} \times \log \left(\frac{\text{observed}}{\text{fitted}} \right).$$

Goodness of fit: Model comparison using the deviance

$$G^2 = 2 \sum_{i,j} \text{observed} \times \log\left(\frac{\text{observed}}{\text{fitted}}\right)$$

It can be shown that G^2 is actually equal to the *deviance* of the model:

$$G^2 = D = 2(L_{\text{saturated}} - L_{\text{model}})$$

Assuming our model fits well enough, then

$$G^2 \sim \chi^2(n-k),$$

where

- $n =$ sample size
- $k =$ number of model parameters.

Goodness of fit: Model comparison using the deviance

		Marijuana Use	
Race	Gender	Yes	No
White	Female	420	620
	Male	483	579
Other	Female	25	55
	Male	32	62

Predictors: *Race* and *Gender* (both factors with 2 levels).
Outcome: *Marijuana Use*, binary (0 = no; 1 = yes).

Goodness of fit: Model comparison using the deviance

```r
# Marijuana data:
mar.use <- data.frame(Race   = c("white", "white", "other", "other"),
                      Gender = c("female", "male", "female", "male"),
                      Yes    = c(420, 483, 25, 32),
                      No     = c(620, 579, 55, 62))

# Logistic regression:
mar.use.fit <- glm(Yes / (Yes + No) ~ Race + Gender,
                   weights = Yes + No,
                   family  = binomial,
                   data    = mar.use)

# Compute G^2 in two different ways:
deviance(mar.use.fit)
sum( residuals(mar.use.fit, type = "deviance")^2 )
```

Output:

```
[1] 0.05798151
---------------
[1] 0.05798151
```

Goodness of fit: Model comparison using the deviance

The chi square test then compares our model to the saturated model:

```
# Saturated model:
mar.use.fit2 <- glm(Yes / (Yes + No) ~ diag(4), # one parameter
                                                # per observation
                    weights = Yes + No,
                    family  = binomial,
                    data    = mar.use)

anova(mar.use.fit, mar.use.fit2, test = "LRT")
```

Output:

```
-----------------------------------------
2         0   0.000000  1 0.057982   0.8097
-----------------------------------------
```

Conclusion:
$\chi^2(1) = .058$, $p = .81$: Our model does not fit significantly worse than the saturated model (a good thing).

Exercise 10

Use the data from Exercise 7-3 (MBTI) and answer the following questions.

1. Compute the AIC for the Models M_1 and M_2, respectively.
2. Compare the two models, M_1 and M_2, using the AICs calculated from 1. Which model is preferable?
3. Using backward elimination, determine which of the 4 factors should be selected in the final model.

Next lecture

Chapter 6.1 and 6.2 will be covered during the next lecture.

However, the below section will be skipped

- From 6.1: 6.1.4-6.1.5, 6.1.6-6.1.7.
- From 6.2: 6.2.4-6.2.6.

Exercise 10 (Japanese/日本語)

Exercise 7-3 (MBTI) のデータを用いて以下の質問に答えよ.

1. モデル M_1 と M_2 の AIC をそれぞれ求めよ.
2. 1. で求めた AIC に基づいてモデル M_1 と M_2 を比較せよ. どちらのモデルがより好ましいでしょうか?
3. 後方消去法 (backward elimination) を用いて 4 つの説明カテゴリーのうち最終モデルに含むべき説明変数を選択せよ.