Data-analysis and Retrieval
Classification: Additional Slides

Ad Feelders

Universiteit Utrecht

Expectation

The expected value of discrete random variable Y is:

$$E(Y) = \sum_{y} y \; P(Y = y)$$

Since $Y \in \{0, 1\}$ we get:

$$E(Y \mid X) = 1 \times P(Y = 1 \mid X) + 0 \times P(Y = 0 \mid X) = P(Y = 1 \mid X)$$

Logit Transformation

Logistic regression assumption:

$$P(Y = 1 \mid X) = \frac{e^{\beta_0 + \beta_1 X}}{1 + e^{\beta_0 + \beta_1 X}}$$

Therefore

$$P(Y = 0 \mid X) = 1 - P(Y = 1 \mid X) = \frac{1}{1 + e^{\beta_0 + \beta_1 X}},$$

and hence the odds are

$$\frac{P(Y = 1 \mid X)}{P(Y = 0 \mid X)} = e^{\beta_0 + \beta_1 X}$$

Finally, the log-odds are

$$\ln\left(\frac{P(Y = 1 \mid X)}{P(Y = 0 \mid X)}\right) = \beta_0 + \beta_1 X$$

Decision boundary and classification rule

Classes are equally likely when

$$\frac{P(Y = 1 \mid X)}{P(Y = 0 \mid X)} = 1$$

and hence

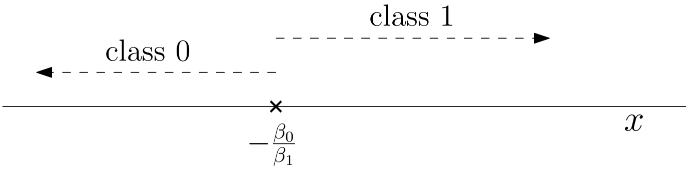
$$\ln\left(\frac{P(Y = 1 \mid X)}{P(Y = 0 \mid X)}\right) = 0$$

So the decision boundary is

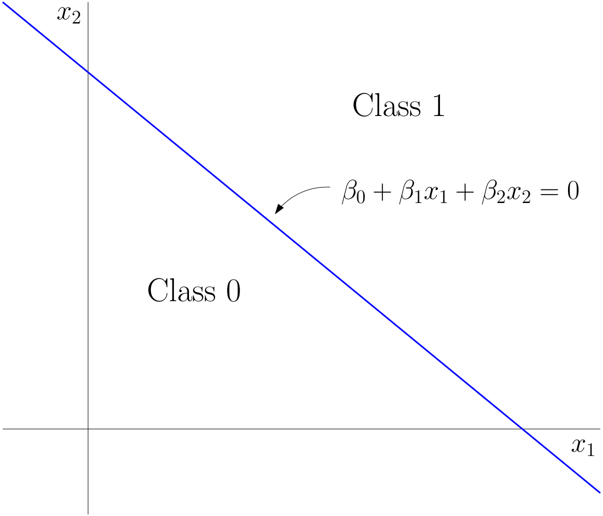
$$\beta_0 + \beta_1 X = 0$$

Assign to class 1 if $\beta_0 + \beta_1 X > 0$ and to class 0 otherwise.
If $\beta_1 > 0$: Assign to class 1 if $X > -\frac{\beta_0}{\beta_1}$ and to class 0 otherwise.
If $\beta_1 < 0$: Assign to class 1 if $X < -\frac{\beta_0}{\beta_1}$ and to class 0 otherwise.

Linear Decision Boundary (one predictor, $\beta_1 > 0$)



Linear Decision Boundary (two predictors)



Maximum Likelihood Estimation: Coin Tossing

$Y = 1$ if heads, $Y = 0$ if tails. $p = P(Y = 1)$.

In a sequence of 10 coin flips we observe
$\mathbf{y} = (1, 0, 1, 1, 0, 1, 1, 1, 1, 0)$.

The likelihood function is

$$\begin{aligned} P(\mathbf{y}|p) &= p \cdot (1-p) \cdot p \cdot p \cdot (1-p) \cdot p \cdot p \cdot p \cdot p \cdot (1-p) \\ &= p^7(1-p)^3 \end{aligned}$$

The corresponding log-likelihood function is

$$\ln P(\mathbf{y}|p) = \ln(p^7(1-p)^3) = 7 \ln p + 3 \ln(1-p)$$

Computing the maximum

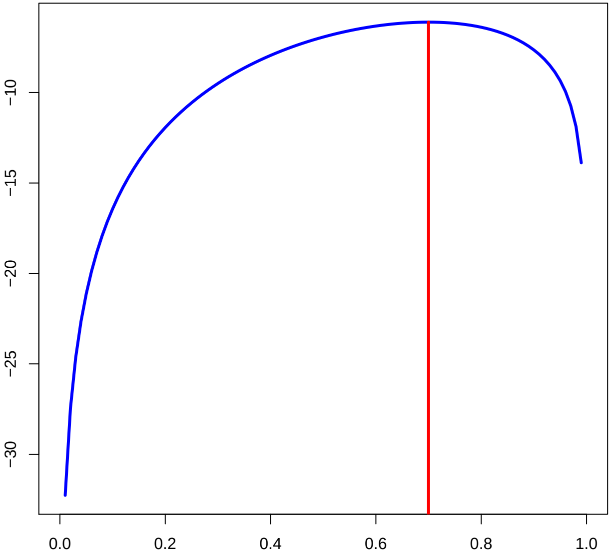
To determine the maximum we take the derivative and equate it to zero (note that $\frac{d \ln x}{dx} = \frac{1}{x}$)

$$\frac{d \ln P(\mathbf{y}|p)}{dp} = \frac{7}{p} - \frac{3}{1-p} = 0$$

which yields maximum likelihood estimate $\hat{p} = 0.7$.

This is just the relative frequency of heads in the sample.

Log-likelihood function for $\mathbf{y} = (1, 0, 1, 1, 0, 1, 1, 1, 1, 0)$



ML estimation for logistic regression

Logistic regression is a bit like the coin tossing example, except that now the probability of success depends on x_i:

$$\begin{aligned} p(x_i) &= P(Y_i = 1 \mid x_i) = \frac{e^{\beta_0 + \beta_1 x_i}}{1 + e^{\beta_0 + \beta_1 x_i}} \\ 1 - p(x_i) &= P(Y_i = 0 \mid x_i) = \frac{1}{1 + e^{\beta_0 + \beta_1 x_i}} \end{aligned}$$

ML estimation for logistic regression

Example

i	x_i	y_i
1	8	0
2	12	0
3	15	1
4	10	1

Likelihood function:

$$\left(\frac{1}{1+e^{\beta_0+8\beta_1}}\right)\left(\frac{1}{1+e^{\beta_0+12\beta_1}}\right)\left(\frac{e^{\beta_0+15\beta_1}}{1+e^{\beta_0+15\beta_1}}\right)\left(\frac{e^{\beta_0+10\beta_1}}{1+e^{\beta_0+10\beta_1}}\right)$$

Unlike with linear regression there is no closed-form solution for the maximum likelihood estimates in logistic regression.

Interpretation

We have

$$\ln\left\{\frac{\hat{P}(Y=1\mid x)}{\hat{P}(Y=0\mid x)}\right\} = -10.6513 + 0.0055x,$$

so with every additional 100 dollars we owe, the log odds increase with $100 \times 0.0055 = 0.55$.

The odds are multiplied by $e^{0.55} \approx 1.73$ so with every additional 100 dollars we owe, the odds increase with 73%.

When x increases with one unit, the odds are multiplied by e^{β_1} because:

$$e^{\beta_0+\beta_1(x+1)} = e^{\beta_0+\beta_1 x+\beta_1} = e^{\beta_0+\beta_1 x} \times e^{\beta_1},$$

since $e^{a+b} = e^a \times e^b$.

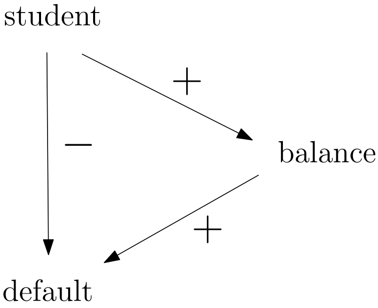
Interpretation

Note that the effect of an increase in balance on the probability of default depends on the value of balance:

- An increase from 1000 to 1100 dollars leads to an increase of the probability of default from 0.006 to 0.01.
- An increase from 1900 to 2000 dollars leads to an increase of the probability of default from 0.45 to 0.59.

The effect depends on where we are on the S-curve.

Confounding



If `balance` is not included as a predictor, then the indirect influence of `student` on `default` via `balance` is attributed to `student`.

If `balance` is included as a predictor as well, then the effects of `student` and `balance` on `default` are separated from each other.

Example: Cushing's Syndrome

Hypertensive disorder associated with over-secretion of cortisol by the adrenal gland. The observations are urinary excretion rates of two steroid metabolites.

The Cushings data frame (in library MASS) has 27 rows and 3 columns:

- Tetrahydrocortisone: urinary excretion rate (mg/24hr).
- Pregnanetriol: urinary excretion rate (mg/24hr).
- Type: underlying type of syndrome
  - a (adenoma)
  - b (bilateral hyperplasia)
  - c (carcinoma)
  - u for unknown (not used in fitting models)

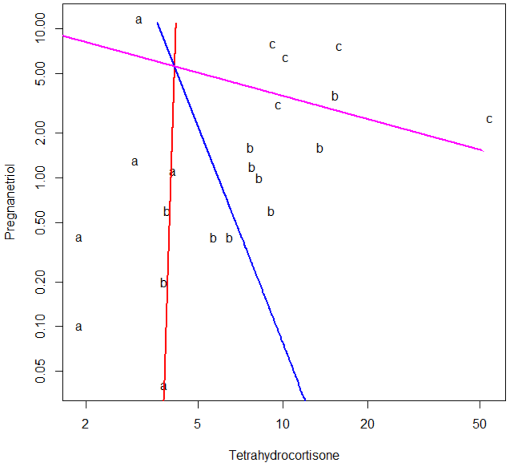
Fitting a Multinomial Logit Model in R

```
> library(MASS)
> library(nnet)
> data(Cushings)
> mycush <- Cushings
> dimnames(mycush)[[2]] <- c("Tetra","Preg","Type")
> cush.multinom <- multinom(Type~log(Tetra)+log(Preg),
                     data=mycush[1:21,],maxit=500)
# weights:  12 (6 variable)
initial  value 23.070858
iter  10 value 6.623970
iter  20 value 6.214841
iter  30 value 6.182968
iter  40 value 6.172650
iter  50 value 6.167699
iter  60 value 6.162723
iter  70 value 6.156685
iter  80 value 6.155298
iter  90 value 6.153807
iter 100 value 6.152597
iter 110 value 6.152041
iter 120 value 6.151229
final  value 6.151167
converged
```

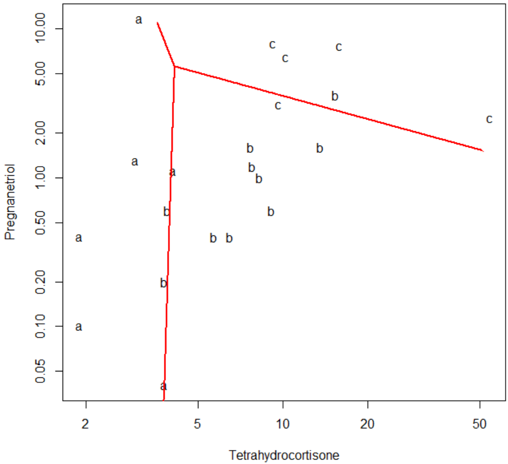
Multinomial Logit: The Fitted Model

```
> summary(cush.multinom)
Call:
multinom(formula = Type ~ log(Tetra) + log(Preg), data = mycush[1:21,
    ], maxit = 500)

Coefficients:
  (Intercept) log(Tetra)  log(Preg)
b   -19.99566   14.37357 -0.2450327
c   -28.83773   16.23923  3.3561273

Std. Errors:
  (Intercept) log(Tetra) log(Preg)
b    18.43773   13.70268 0.6691037
c    18.71154   13.35303 2.0981000

Residual Deviance: 12.30233
AIC: 24.30233
```

Decision Boundary: a-b (red), a-c (blue), b-c (purple)



Decision Boundary



Prediction and Confusion Matrix

```
> cush.pred <- predict(cush.multinom,mycush[1:21,],type="class")
> cush.confmat <- table(mycush[1:21,3],cush.pred)
> cush.confmat
   cush.pred
    a b c
  a 5 1 0
  b 2 7 1
  c 0 1 4
  u 0 0 0

# Accuracy
> 16/21
[1] 0.7619048
```