SMIII lectures

Week 7

Estimation

Estimation

For the logistic regression model, the parameters and approximate standard errors are obtained using the method of maximum likelihood.

- Recall for a scalar parameter, θ, the maximum likelihood estimate $\hat{\theta}$ is obtained by maximizing the log-likelihood function $\ell(\theta; \boldsymbol{y})$ and,
- for large n,
$$\text{var}(\hat{\theta}) \approx \frac{1}{\mathcal{I}(\theta)}$$
where
$$\mathcal{I}(\theta) = E\left(-\frac{\partial^2 \ell}{\partial \theta^2}\right).$$

In practice, an approximate variance is obtained from $1/\mathcal{I}(\hat{\theta})$.

Maximum likelihood

- The method can also be generalised to the case of a vector parameter $\boldsymbol{\theta} \in \mathbb{R}^p$.
- In particular, the maximum likelihood estimate $\hat{\boldsymbol{\theta}}$ is the vector that minimizes the log-likelihood function, $\ell(\boldsymbol{\theta}; \boldsymbol{y})$ and
- for large n,
$$\text{Var}(\hat{\boldsymbol{\theta}}) \approx [\mathcal{I}(\boldsymbol{\theta})]^{-1}$$
where $\mathcal{I}(\boldsymbol{\theta})$ is the $p \times p$ matrix
$$E\left(-\frac{\partial^2 \ell}{\partial \theta_j \partial \theta_k}\right).$$

The log-likelihood function

Consider now the logistic regression model and the binomial likelihood,

$$p(\boldsymbol{y}; \boldsymbol{\beta}) = \prod_{i=1}^{m} \binom{n_i}{y_i} \pi_i^{y_i} (1 - \pi_i)^{n_i - y_i}.$$

The log-likelihood is thus

$$\begin{aligned}
\ell(\boldsymbol{\beta}; \boldsymbol{y}) &= \sum_{i=1}^{m} y_i \log \pi_i + \sum_{i=1}^{m} (n_i - y_i) \log(1 - \pi_i) + \log\left(\prod_{i=1}^{m} \binom{n_i}{y_i}\right) \\
&= \sum_{i=1}^{m} \left(y_i \log\left(\frac{\pi_i}{1 - \pi_i}\right) + n_i \log(1 - \pi_i)\right) + \log\left(\prod_{i=1}^{m} \binom{n_i}{y_i}\right) \\
&= \sum_{i=1}^{m} \left(y_i \eta_i - n_i \log(1 + e^{\eta_i})\right) + \text{const.}
\end{aligned}$$

The maximum likelihood equations

To find the maximum likelihood estimates, we need to solve the equations

$$\frac{\partial \ell}{\partial \beta_j} = 0 \text{ for } j = 1, 2, \ldots, p.$$

The maximum likelihood equations

Now,

$$\frac{\partial \ell}{\partial \beta_j} = \sum_{i=1}^{m} \frac{\partial \ell}{\partial \eta_i} \frac{\partial \eta_i}{\partial \beta_j}$$

and noting that

$$\frac{\partial}{\partial \eta_i} \log(1 + e^{\eta_i}) = \frac{e^{\eta_i}}{1 + e^{\eta_i}} = \pi_i$$

we find

$$\frac{\partial \ell}{\partial \beta_j} = \sum_{i=1}^{m} (y_i - n_i \pi_i) \frac{\partial \eta_i}{\partial \beta_j}.$$

The maximum likelihood equations

Using vector notation, and noting that

$$\boldsymbol{\eta} = X\boldsymbol{\beta} \Rightarrow \left[\frac{\partial \eta_i}{\partial \beta_j} \right] = X$$

we find the score vector, expressed as column vector, is

$$S(\boldsymbol{\beta}) = \left[\frac{\partial \ell}{\partial \beta_j} \right] = X^T(\boldsymbol{y} - \boldsymbol{\mu})$$

where $\boldsymbol{\mu}$ is the vector of expected frequencies

$$\mu_i = n_i \pi_i \text{ with } \pi_i = \frac{e^{\eta_i}}{1 + e^{\eta_i}}.$$

The informaton matrix

To obtain the Fisher information matrix, we use the fact that

$$\text{Var}(S(\boldsymbol{\beta})) = \mathcal{I}(\boldsymbol{\beta})$$

where

$$S(\boldsymbol{\beta}) = X^T(\boldsymbol{Y} - \boldsymbol{\mu})$$

is the score, expressed as a column vector.

The information matrix

Since X is a fixed matrix, we obtain

$$\begin{aligned}\mathcal{I}(\boldsymbol{\beta}) &= \operatorname{Var}(X^T(\boldsymbol{Y}-\boldsymbol{\mu})) \\ &= X^T \operatorname{Var}(\boldsymbol{Y})X \\ &= X^T D X\end{aligned}$$

where

$$\begin{aligned}D &= \operatorname{diag}(n_1\pi_1(1-\pi_1), n_2\pi_2(1-\pi_2), \ldots, n_m\pi_m(1-\pi_m)) \\ &= \operatorname{Var}(\boldsymbol{Y})\end{aligned}$$

since $Y_1, Y_2, \ldots, Y_m$ are assumed independent

$$Y_i \sim B(n_i, \pi_i)$$

MLE

This system cannot be solved analytically, and an iterative scheme such as the Newton-Raphson algorithm or the Fisher scoring algorithm is usually used.

Newton-Raphson algorithm

The Newton-Raphson algorithm consists of choosing an initial estimate $\boldsymbol{\beta}^{(0)}$ and then iterating

$$\boldsymbol{\beta}^{(t+1)} = \boldsymbol{\beta}^{(t)} - \left[S'(\boldsymbol{\beta}^{(t)})\right]^{-1} S(\boldsymbol{\beta}^{(t)})$$

where

$$S'(\boldsymbol{\beta})$$

is the $p \times p$ matrix

$$\left[\frac{\partial S_j}{\partial \beta_k}\right] = \left[\frac{\partial^2 \ell}{\partial \beta_j \partial \beta_k}\right].$$

Fisher Scoring

The Fisher scoring algorithm is obtained by replacing the matrix $S'(\boldsymbol{\beta})$ by its expected value, namely $-\mathcal{I}(\boldsymbol{\beta})$.

Thus the iterative step of the Fisher scoring algorithm in the general case is

$$\boldsymbol{\beta}^{(t+1)} = \boldsymbol{\beta}^{(t)} + [\mathcal{I}(\boldsymbol{\beta}^{(t)})]^{-1} S(\boldsymbol{\beta}^{(t)}).$$

Fisher Scoring

For the logistic regression model, this can be simplified to

$$\boldsymbol{\beta}^{(t+1)} = \boldsymbol{\beta}^{(t)} + (X^T D^{(t)} X)^{-1} X^T (\boldsymbol{y} - \boldsymbol{\mu}^{(t)})$$

where $D^{(t)}$ and $\boldsymbol{\mu}^{(t)}$ are

- $D = \text{diag}(n_1\pi_1(1-\pi_1), n_2\pi_2(1-\pi_2), \ldots, n_m\pi_m(1-\pi_m))$
- $\boldsymbol{\mu} = (n_1\pi_1, n_2\pi_2, \ldots, n_m\pi_m)^T$

evaluated at $\boldsymbol{\beta} = \boldsymbol{\beta}^{(t)}$.

Initial value

The initial approximation, $\boldsymbol{\beta}^{(0)}$ is usually taken to be

$$\boldsymbol{\beta}^{(0)} = (X^TX)^{-1}X^T\boldsymbol{v}$$

where

$$v_i = \log\left(\frac{y_i + 0.5}{n_i - y_i + 0.5}\right).$$

Inference for regression coefficients

For large m or large n_i, the approximate distribution of $\hat{\beta}_j$ is

$$N\left(\beta_j, \sqrt{[\mathcal{I}(\hat{\boldsymbol{\beta}})^{-1}]_{jj}}\right)$$

where

$$[\mathcal{I}(\hat{\boldsymbol{\beta}})^{-1}]_{jj}$$

is the jth diagonal element of

$$\mathcal{I}(\hat{\boldsymbol{\beta}})^{-1}$$

Confidence interval

The approximate $100(1-\alpha)\%$ confidence intervals for β_j is

$$\hat{\beta}_j \pm z_{\alpha/2}\sqrt{[\mathcal{I}(\hat{\boldsymbol{\beta}})^{-1}]_{jj}}.$$

Hypotheses concerning several parameters

Consider the logistic regression model

$$M: \; \boldsymbol{\eta} = X\boldsymbol{\beta} \text{ where } \boldsymbol{\beta} = (\beta_1, \beta_2, \ldots, \beta_p)^T$$

and the hypothesis

$$H_0: \; \beta_{p_0+1} = \beta_{p_0+2} = \ldots = \beta_{p-1} = \beta_p = 0.$$

The log likelihood ratio test statistic is defined by

$$G^2 = 2(\ell(\hat{\boldsymbol{\beta}}) - \ell(\hat{\boldsymbol{\beta}}_0))$$

where $\hat{\boldsymbol{\beta}}$ and $\hat{\boldsymbol{\beta}}_0$ are the MLEs under M and H_0 respectively.

If H_0 is true then the asymptotic distribution of G^2 is $\chi^2_{p-p_0}$ and we reject H_0 for

$$G^2 \geq \chi^2_{p-p_0,\alpha}.$$

Deviance statistic

The deviance statistic for any model M is defined by

$$D(M) = 2(\hat{\ell} - \ell(\hat{\boldsymbol{\beta}}))$$

where $\hat{\ell}$ is the value of the likelihood maximized without restriction and is produced in R as the **Residual Deviance**.

For grouped data such that $n_i\pi_i(1-\pi_i)$ are all large, the asymptotic distribution of the residual deviance is χ^2_{m-p}. In this case, the residual deviance can be used to test the overall fit of the model.

Model fit

As with any statistical analysis we need to check the model assumptions.

In this case, there are two approaches that can be used.

- A formal approach: hypothesis testing.
- An less formal approach: observe residuals from the model fit.

Formal approach

The model in question is embedded within a more general model and a hypothesis test is conducted.

We have previously seen that the likelihood ratio test statistic for a test of the given model against the saturated model can be obtained as the residual deviance.

The likelihood ratio test statistic can also be obtained using the `anova()` function to compare the glm object for the model `M1` against a more general model `M2`.

When using the `anova()` function in R to compare two models, it is essential that one of the models is a sub-model of the other.

Less formal approach

- Plot (suitably defined) residuals from the model fit.

Pearson residuals

The Pearson residuals:

$$r_i^{(p)} = \frac{y_i - n_i\hat{\pi}_i}{\sqrt{n_i\hat{\pi}_i(1-\hat{\pi}_i)}};$$

The deviance residuals

The Deviance residuals:

$$r_i^{(d)} = \operatorname{sgn}(y_i - n_i\hat{\pi}_i)\sqrt{2y_i \log\frac{y_i}{n_i\hat{\pi}_i} + 2(n_i - y_i)\log\frac{n_i - y_i}{n_i(1-\hat{\pi}_i)}}$$

The Pearson residuals may be interpreted as the contributions to the Pearson χ^2 statistic and the deviance residuals are the corresponding contributions to the residual deviance.

Residuals

Both types of residuals can be used in roughly the same way as residuals from ordinary linear regression.

In particular, these residuals may be plotted against the fitted values and also against the individual predictor variables.

In considering these residuals, some care must be taken as they are not standardized.

Non-grouped data residuals

The preceding residuals are most useful when dealing with grouped data.

For ungrouped data, where every y-value is either 1 or 0, the residuals can take only two possible values for any given $\hat{\pi}$.

This will produce artifacts that can make the residual plot difficult to interpret.

Standardized residuals can also be calculated using approximate variances, but the details are beyond the scope of this course.

Approximate influence diagnostics are also available but are beyond the scope of this course.

Over-dispersion

It should be noted that the assumption of binomial variability in the binary outcomes may be violated in various ways.

For example, the binomial distribution dictates that

- $E(Y_i) = n_i\pi_i$, and
- $\operatorname{var}(Y_i) = n_i\pi_i(1-\pi_i)$.

When $\operatorname{var}(Y_i) > n_i\pi_i(1-\pi_i)$ the distribution is said to be over-dispersed (relative to the binomial).

When $\operatorname{var}(Y_i) < n_i\pi_i(1-\pi_i)$ the distribution is said to be under-dispersed (relative to the binomial).

Existence of MLE

```r
M1 <- glm(y ~ x, data = df, family = binomial)
```

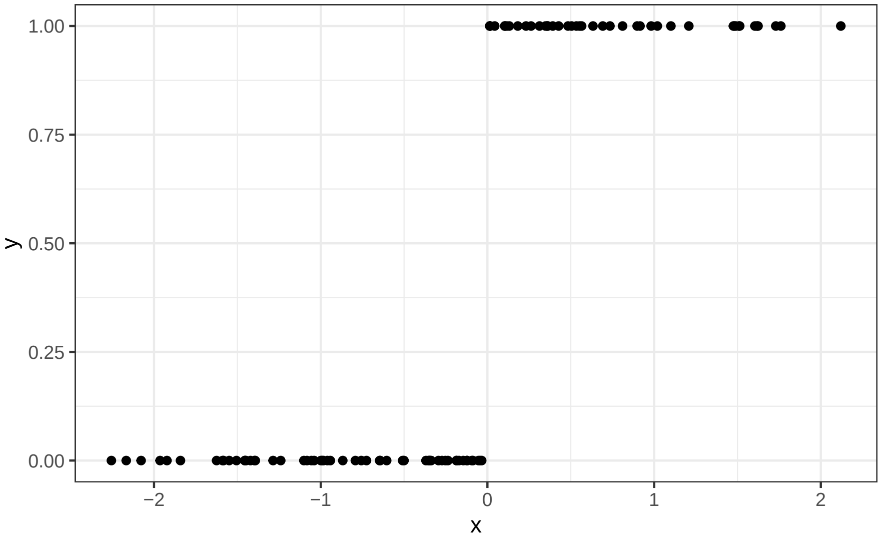
```
## Warning: glm.fit: algorithm did not converge
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
summary(M1)
```

```
##
## Call:
## glm(formula = y ~ x, family = binomial, data = df)
##
## Deviance Residuals:
##        Min          1Q      Median          3Q         Max
## -2.821e-04  -2.100e-08  -2.100e-08   2.100e-08   2.060e-04
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)     7.741   2481.570   0.003    0.998
## x             720.837  91641.431   0.008    0.994
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1.3666e+02  on 99  degrees of freedom
## Residual deviance: 1.4814e-07  on 98  degrees of freedom
## AIC: 4
##
## Number of Fisher Scoring iterations: 25
```

Plot to explain



Warning signs

When using the logistic regression, it is good practice to check for convergence.

In particular,

- Warning messages (may not always be present)
- High number of iterations.
- Some parameter estimates and standard errors with very large numerical values.
- Estimated probabilities very close to zero or one.

Implementation in R

Setup

Consider an intervention in which we break a class into groups, and have each group take the same test at a single time: 1pm, 3pm, 7pm or 11pm. After the test, we record the number of students that passed the test.

Here is the data:

time	n	y
1	21	19
3	23	19
7	19	15
11	12	6

Let us

- find X for logistic regression with intercept and slope (standard)
- find $\boldsymbol{\beta}^{(0)}$
- find $\boldsymbol{\beta}^{(t+1)}$ given $\boldsymbol{\beta}^{(t)}$

find X

```
x = c(1,3,7,11) #time data - numeric predictor
X <- matrix(1,nrow = length(x), ncol = 2) #design matrix
X[,2] <- x
X
```

```
##      [,1] [,2]
## [1,]    1    1
## [2,]    1    3
## [3,]    1    7
## [4,]    1   11
```

find $\boldsymbol{\beta}^{(0)}$

get rest of data:

```
n = c(21,23,19,12)
y = c(19,19,15,6)
```

construct $\boldsymbol{\beta}^{(0)} = (X^T X)^{-1} X^T \boldsymbol{v}$

```
v <- log((y + 0.5)/(n-y+0.5))
b0 = solve(t(X) %*% X) %*% t(X) %*% v

b0 #initial estimate of beta
```

```
##            [,1]
## [1,]  2.2197873
## [2,] -0.1873603
```

find $\boldsymbol{\beta}^{(t+1)}$ given $\boldsymbol{\beta}^{(t)}$

Here I find $\boldsymbol{\beta}^{(1)}$ using $\boldsymbol{\beta}^{(0)}$. Recall

$$\boldsymbol{\beta}^{(t+1)} = \boldsymbol{\beta}^{(t)} + (X^T D^{(t)} X)^{-1} X^T (\boldsymbol{y} - \boldsymbol{\mu}^{(t)})$$

We need $\boldsymbol{\mu}^{(t)}$ and $D^{(t)}$, both functions of

```
eta = X %*% b0
eta
```

```
##           [,1]
## [1,] 2.0324270
## [2,] 1.6577065
## [3,] 0.9082655
## [4,] 0.1588244
```

Using $\eta^{(t)}$ get $n_i\pi_i^{(t)}$,

```
pii <- exp(eta)/(1+exp(eta))
npii = n*pii
npii
```

```
##           [,1]
## [1,] 18.567358
## [2,] 19.318387
## [3,] 13.540257
## [4,]  6.475474
```

and $D^{(t)}$

```
D <- matrix(0,nrow=4,ncol=4)
diag(D) <- npii*(1-pii)
D
```

```
##          [,1]     [,2]     [,3]    [,4]
## [1,] 2.150845 0.000000 0.000000 0.00000
## [2,] 0.000000 3.092296 0.000000 0.00000
## [3,] 0.000000 0.000000 3.890859 0.00000
## [4,] 0.000000 0.000000 0.000000 2.98116
```

Finally

```
bt1 <- b0 + solve(t(X) %*% D %*% X) %*% t(X) %*% (y - npii)
bt1
```

```
##            [,1]
## [1,]  2.3847421
## [2,] -0.1999544
```