Ch. 11 Logistic Regression

- The Model
- Interpretation of the Parameters
- Parameter Estimation
- Inference
- Model Checking
Generalized Linear Models

- Many data sets do not satisfy the normality and constant variance assumptions. In the past, data analysts treated this problem with clever transformations of the response variable which would make it behave more like a normal random variable with a constant variance.

- However, a more direct approach is available, if a reasonable model can be constructed. The parameters for the model can be estimated by maximum likelihood.
11.1 The Logistic Regression Model

- Data: Responses ($y_i$’s) are categorical variables with 2 levels (coded as 1 and 0). Predictor values ($x_i$’s) can be categorical and/or numeric.

- For $i = 1, 2, \ldots, n$,

\[
P(y_i = 1) = \pi_i
\]

\[
P(y_i = 0) = 1 - \pi_i
\]

where $\pi_i$ are probabilities (values are between 0 and 1).

- Such data may come in the form of a two-way contingency table (see section 11.2.1)

- We are interested in modelling $\pi_i$ in terms of $x_i$:

\[
\pi_i = \pi(x_i).
\]
Another possibility: 
$y_i$ is a Binomial random variable, whose proportion parameter $p$ depends on the predictor variable $x_i$:

$$y_i \sim \text{bin}(n_i, p(x_i))$$

$$E[y|x] = n_i p(x_i)$$

$$V(y|x) = n_i p(x_i)(1 - p(x_i))$$

If $p(x)$ is not constant, then the variance cannot be constant.
The Link Function

There is a function $g(z)$, called the link function which satisfies:

$$g(\pi(x)) = \beta_0 + \beta_1 x$$

We will consider 2 possible link functions:

**identity:** $g(p) = p$

**logit:** $g(p) = \log\left(\frac{p}{1-p}\right)$

Others exist. The logit is preferred.
11.2 Interpretation of the Logit Link

- Odds:

\[
\frac{\pi(x)}{1 - \pi(x)}
\]

This is the ratio of the probability of occurrence of an event to the probability of it not occurring.

Odds can take on any value between 0 and infinity.

e.g. Suppose \( \pi(x) = .75 \). Then the odds of the event is \( .75/.25 = 3 \).

e.g. If the odds of an event is 2, then, you can find that the probability of the event is \( 2/(2+1) = 2/3 \).

i.e.

\[
\pi(x) = \frac{\text{odds}(x)}{\text{odds}(x) + 1}.
\]
Interpretation of the Logit Link

- **Log Odds:**

\[
\log \left( \frac{\pi(x)}{1 - \pi(x)} \right)
\]

This is the logit link function. Its main advantage is that it can take values on the entire real line.

- Thus, we can equate it to \( \beta_0 + \beta_1 x \) for any given \( \beta_0 \) and \( \beta_1 \), and \( x \) does not have to be restricted in any way. We will always obtain valid log odds.

Using the identity link leads to restrictions on \( x \) so that valid probabilities will be obtained.
Interpretation of the Logit Link

• Consider

\[
\log \left( \frac{\pi(x)}{1 - \pi(x)} \right) = \beta_0 + \beta_1 x
\]

• Then

\[
\beta = \log \left( \frac{\pi(x + 1)}{1 - \pi(x + 1)} \right) - \log \left( \frac{\pi(x)}{1 - \pi(x)} \right).
\]

• Thus, \( \beta \) is the log of the odds ratio for the occurrence of the event at \( x + 1 \) to the occurrence of the event at \( x \).
11.3 Parameter Estimation

- **Data:**
  
  \[
  x_1 \quad n_1 \quad y_1 \\
  x_2 \quad n_2 \quad y_2 \\
  \ldots \quad \ldots \quad \ldots \\
  x_m \quad n_m \quad y_m \\
  \]

- **Likelihood:**
  
  \[
  \prod_{i=1}^{m} P(Y_i = y_i) = \prod_{i=1}^{m} \binom{n_i}{y_i} p^y_i(1 - p)^{n_i-y_i}
  \]
Example

- \( x_i = 0 \) for \( i = 1, 2, 3, 4, 5; \)
  \( x_i = 1 \) for \( i = 6, 7, 8, 9, 10. \)

- **Identity link function:** \( p(x_i) = \beta_0 + \beta_1 x_i. \)

- **Log Likelihood:**

\[
\ell(\beta_0, \beta_1) = \sum_{i=1}^{10} \log \left( \frac{n_i}{y_i} \right) + \sum_{i=1}^{5} y_i \log(\beta_0) \\
+ \sum_{i=1}^{5} (n_i - y_i) \log(1 - \beta_0) \\
+ \sum_{i=6}^{10} y_i \log(\beta_0 + \beta_1) \\
+ \sum_{i=6}^{10} (n_i - y_i) \log(1 - \beta_0 - \beta_1)
\]
Differentiating the log likelihood with respect to $\beta_0$ and $\beta_1$ and setting the derivatives to 0, gives the estimators:

\[ \hat{\beta}_0 = \frac{\sum_{i=1}^{5} y_i}{\sum_{i=1}^{5} n_i} \]

and

\[ \hat{\beta}_1 = \frac{\sum_{i=6}^{10} y_i}{\sum_{i=6}^{10} n_i} \]
Example (Cont’d)

• Fitting the model with `glm()`:

  ○ Enter the data into a dataframe:

    ```
    > x <- c(rep(0,5), rep(1,5))
    > n <- c(3,6,4,8,10,2,5,7,9,8)
    > y <- c(1,2,2,2,3,1,3,4,7,7)
    > nxy <- data.frame(n, x, y)
    ```

  ○ Fit the model:

    ```
    > nxy.glm <- glm(I(y/n) ~ x, weights=n, data=nxy,
                   family=binomial(link="identity"))
    ```
Example (Cont’d)

- **Obtain the parameter estimates:**

  ```
  > summary(nxy.glm)
  Coefficients:
      Estimate Std. Error  z value Pr(>|z|)
(Intercept) 0.32258    0.08396   3.842   0.000122 ***
x          0.38710    0.11703   3.308   0.000941 ***
  ```

- **The fitted model:**

  \[
  \hat{p}(x) = 0.323 + 0.387x
  \]

  This means that the probability of a “success” is 0.323 when \( x = 0 \) and 0.710 when \( x = 1 \).
Example (Cont’d)

- For this simple model, we could also use our formulas directly to calculate the coefficients:

```r
> sum(y[1:5])/sum(n[1:5])
[1] 0.3225806
> sum(y[6:10])/sum(n[6:10])
[1] 0.7096774
> sum(y[6:10])/sum(n[6:10])-.3225806
[1] 0.3870968
```
Example (Cont’d)

- You can use the predict function to compute fitted values:

```r
> predict(nxy.glm, newdata =
  data.frame(x = c(0, 1), n = c(10,15)))
  1    2
0.3225806  0.7096774

... a bit boring in this case, since there are only 2 possibilities ...
• **Using the logit link:**
  
  \[ \log \left( \frac{p}{1-p} \right) = \beta_0 + \beta_1 x \]
  
  or
  
  \[ p = \frac{e^{\beta_0 + \beta_1 x}}{1 + e^{\beta_0 + \beta_1 x}} \]
  
  • The above expression can be plugged into the likelihood function.
  
  • Again, the likelihood can be maximized with respect to \( \beta_0 \) and \( \beta_1 \) to get parameter estimates. Formulas can be obtained without difficulty in this case, and this is left as an exercise.
Example (Cont’d)

- **Fitting the model with `glm()`**:

```r
> nxy.glm <- glm(I(y/n) ~ x, weights=n, data=nxy,
  family=binomial)
> summary(nxy.glm)
Coefficients:

          Estimate Std. Error   z value  Pr(>|z|)
(Intercept) -0.7419    0.3842  -1.9311   0.05348 .
x          1.6358    0.5515   2.9664   0.00302 **
```

- **The fitted model**:

\[
\log \left( \frac{p}{1 - p} \right) = -0.742 + 1.64x
\]

Note you have to apply the inverse of the logit transformation to get back probabilities.
Beetle Example

- In an experiment (Bliss, 1935), some beetles were exposed to gaseous carbon disulphide at various concentrations for 5 hours. The numbers killed at each dose were recorded.

- beetles.txt contains the following lines:

<table>
<thead>
<tr>
<th>dose</th>
<th>n.insects</th>
<th>n.killed</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.6907</td>
<td>59</td>
<td>6</td>
</tr>
<tr>
<td>1.7552</td>
<td>62</td>
<td>18</td>
</tr>
<tr>
<td>1.8113</td>
<td>63</td>
<td>52</td>
</tr>
<tr>
<td>1.8610</td>
<td>62</td>
<td>61</td>
</tr>
<tr>
<td>1.7242</td>
<td>60</td>
<td>13</td>
</tr>
<tr>
<td>1.7842</td>
<td>56</td>
<td>28</td>
</tr>
<tr>
<td>1.8369</td>
<td>59</td>
<td>53</td>
</tr>
<tr>
<td>1.8839</td>
<td>60</td>
<td>60</td>
</tr>
</tbody>
</table>
Example (Cont’d)

- **Read in the data:**

  ```r
  beetles <- read.table("beetles.txt", header=T)
  ```

- ```R
  > beetles.glm <- glm(I(n.killed/n.insects) ~ dose, data=beetles, 
  family=binomial, weights=n.insects)
  > summary(beetles.glm)
  ```

  Coefficients:

  |                     | Estimate | Std. Error | z value | Pr(>|z|) |
  |---------------------|----------|------------|---------|----------|
  | (Intercept)          | -60.717  | 5.181      | -11.72  | <2e-16   *** |
  | dose                | 34.270   | 2.912      | 11.77   | <2e-16   *** |
Example (Cont’d)

○ The fitted model:

\[
\log \left( \frac{\hat{p}}{1 - \hat{p}} \right) = -60.7 + 34.3x
\]

where \( x \) represents dose.

○ As dose increases, the log odds that the insects will be killed increases, i.e. the probability of death increases with dose.
11.4 Inference

- Standard errors can be calculated for the parameter estimates (Section 11.4.3). These are given in the `glm` output.
- The diagonal elements of the Fisher information matrix are approximations to the coefficient variances.
- The Fisher information matrix is the inverse of the second derivative of the log likelihood.
- Remember that the second derivative is related to how much curvature is in a function.
- If the likelihood function is flat near the parameter estimate, then the second derivative will be small, so its inverse will be large $\Rightarrow$ large variance.
- If the likelihood function has a fairly sharp peak at the parameter estimate, then the second derivative will be large, so its inverse will be small $\Rightarrow$ small variance.
11.4 Inference

- Another way to calculate standard errors is to use bootstrap resampling.
- The basic idea is to treat the sample estimates as if they were the population parameters, and to simulate from this “population”.
- Parameter estimates can be recalculated from the simulated data.
- Doing this repeatedly gives a set of parameter estimates which should resemble the sampling distribution of the estimator.
- We can calculate the standard deviation of this set of ‘estimates’ and that will be an approximation to the standard error.
Bootstrap Example

- \( n \leftarrow 10000 \)  # very large sample size
  
  \[
  g \leftarrow \text{function}(x, \beta_0, \beta_1) \beta_0 + \beta_1 x
  \]
  
  \[
  l \leftarrow \text{function}(g) \exp(g)/(1+\exp(g))
  \]
  
  \[
  p \leftarrow \text{function}(x, \beta_0, \beta_1) l(g(x, \beta_0, \beta_1))
  \]
  
  \( x \leftarrow \text{seq}(-1.5, 1.5, \text{length}=n) \)  # predictor values
  
  \( \beta_0 \leftarrow -3 \)  # model parameters
  
  \( \beta_1 \leftarrow 3 \)
  
  \( y \leftarrow \text{rbinom}(n, 1, p(x, \beta_0, \beta_1)) \)  # response values
  
  \( y.\text{glm} \leftarrow \text{glm}(y \sim x, \text{family}={}\text{binomial}) \)
Bootstrap Example

- \( \beta_0^{\text{hat}} \leftarrow \text{coef}(y.\text{glm})[1] \) # estimated parameters
- \( \beta_1^{\text{hat}} \leftarrow \text{coef}(y.\text{glm})[2] \)

\>
- \( \beta_0^{\text{hat}} \)
  (Intercept)
  -2.969129
- \( \beta_1^{\text{hat}} \)
  x
  3.01345
Bootstrap Example

- SEs <- summary(y.glm)$coefficients[,2]  # theoretical SEs
  > SEs
  (Intercept)   x
  0.06664972  0.07224331
Bootstrap Example

- beta0hathat <- numeric(1000)
  beta1hathat <- numeric(1000)
  for (i in 1:1000) {
    yhat <- rbinom(n, 1, p(x, beta0hat, beta1hat))
    yhat.glm <- glm(yhat ~ x, family=binomial)
    beta0hathat[i] <- coef(yhat.glm)[1]
    beta1hathat[i] <- coef(yhat.glm)[2]
  }
  > sd(beta0hathat)
  [1] 0.06465081
  > sd(beta1hathat)
  [1] 0.06835065

- Exercise: What are the standard error estimates for the coefficients in the Beetles data set? Compare with bootstrap standard error estimates.
11.5 Model Checking

- Not as straightforward as in the linear regression case, but it is still possible.

- Pearson Residuals (see 11.5.3)

- Deviance Residuals (see 11.5.3)

- Hat Matrix

- Influence Diagnostics