Tutorial 6

STAT3015/4030/7030 Generalised Linear Modelling

The Australian National University

Week 6, 2017

Overview

1. Summary
2. Question 1
3. Question 2

Logistic regression for binary response variables

Assume the response variable, Y, can only take values 0 and 1,

$$Y = \begin{cases} 0 & \text{with lung cancer} \\ 1 & \text{healthy} \end{cases}$$

Let $\mu = Mean\{Y|X_1, \ldots, X_p\}$. We can model this mean response through a link function $g(\mu)$:

$$g(\mu) = \eta = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_p X_p$$

We no longer assume that our Y follows a Gaussian distribution.

Link functions

We need to specify the link function $g(\cdot)$ before we can model the mean of the response variable. There are three commonly used link functions:

1. Logit: $g(p) = \log \frac{p}{1-p}$
2. Probit: $g(p) = \Phi^{-1}(p)$
3. Complementary log-log: $g(p) = \log(-\log(1-p))$

Since we model η by $g(\mu)$, we need to do back transformation to get $\hat{\mu}$.

Logistic regression model

The inverse of the logit function is called the logistic function (or inverse logit):

$$p = \frac{exp(\eta)}{1 + exp(\eta)}$$

Our logistic regression model for binary response is then:

$$g(p) = logit(p) = \log \frac{p}{1-p} = \beta_0 + \beta_1 X_1 + \cdots + \beta_q X_q$$

The response Y is assumed to have a Bernoulli distribution with probability p:

$$Y = \begin{cases} 1 & \text{with probability} \quad p \\ 0 & \text{with probability} \quad 1-p \end{cases}$$

Modelling GLM in R

We use “glm()” function to build GLM models in RStudio. For example, in Q1 (a) we can use the following command:

```
m1 <- glm(FAILURE ~ TEMP, family = binomial(link = "logit"))
```

Modelling GLM in R

We use “glm()” function to build GLM models in RStudio. For example, in Q1 (a) we can use the following command:

```
m1 <- glm(FAILURE ~ TEMP, family = binomial(link = "logit"))
```

Notice here we have used **“family = binomial”** to indicate that we are building a logistic regression model. Later we will also use **Gaussian**, **Poisson** and **Gamma**. (Make sure you are comfortable with these distributions.)

Modelling GLM in R

We use “glm()” function to build GLM models in RStudio. For example, in Q1 (a) we can use the following command:

```
m1 <- glm(FAILURE ~ TEMP, family = binomial(link = "logit"))
```

Notice here we have used **“family = binomial”** to indicate that we are building a logistic regression model. Later we will also use **Gaussian**, **Poisson** and **Gamma**. (Make sure you are comfortable with these distributions.)

We also specify that we are using the “logit” link function. We can change this to “probit” or “cloglog”.

Modelling GLM in R

We use “glm()” function to build GLM models in RStudio. For example, in Q1 (a) we can use the following command:

```
m1 <- glm(FAILURE ~ TEMP, family = binomial(link = "logit"))
```

Notice here we have used **“family = binomial”** to indicate that we are building a logistic regression model. Later we will also use **Gaussian**, **Poisson** and **Gamma**. (Make sure you are comfortable with these distributions.)

We also specify that we are using the “logit” link function. We can change this to “probit” or “cloglog”.

The lecture note says some theoretical work has shown that the logistic model is more robust than the probit model. (Page 30)

Drop in deviance test

After building GLM models, we usually need to select the optimal model (i.e., compare different models). In GLM, we calculate a quantity known as deviance which is a measure of error. Lower deviance means a better fit to the data.

$$deviance = Constant - 2 \times \log(Maximum\ Likelihood)$$

Drop in deviance test

- If a predictor is added that is simply random noise, we expect deviance to decrease by 1 on average.
- When an informative predictor is added, we expect deviance to decrease by more than 1.
- When d predictors are added to a model, we expect deviance to decrease by more than d.

Then we need to determine how much difference from d should be treated as significant. $\Longrightarrow$ We need a distribution for the test.

Drop in deviance test

The Likelihood ratio test we learned before can be expressed as:

$$LRT = deviance_{reduced} - deviance_{full}$$

Deviance values can be found in summary outputs. We still compare the drop-in-deviance result to a χ^2_d distribution, with d denoting the difference in the number of parameters.

Exponential family of probability distributions

Any probability distribution with a density of the following form:

$$f(y;\mu,\phi) = \exp\left\{\frac{y\mu - b(\mu)}{\phi} + c(y,\phi)\right\}$$

for some specified functions $b(\mu), c(\mu)$ and $d(y,\phi)$.

To find $b(\mu), c(\mu)$ and $d(y,\phi)$ of a given pmf or pdf we need to take natural logarithm and then apply exponential function. The function $b(\mu)$ is

often called the **canonical link** function. Here we use binomial and Poisson distributions as examples.

Exponential family

Verify that the function $g(\mu) = \log\{\mu/(n-\mu)\}$ is the canonical link for the *binomial*(n, p) family.

Exponential family

Verify that the function $g(\mu) = \log\{\mu/(n-\mu)\}$ is the canonical link for the *binomial*(n, p) family.

$$\begin{aligned}
f(y; n, p) &= \binom{n}{y} p^y (1-p)^{n-y} \\
&= \exp\left(y \log p + (n-y)\log(1-p) + \log\binom{n}{y} \right) \\
&= \exp\left(y \log \frac{p}{1-p} + n\log(1-p) + \log\binom{n}{y} \right) \\
&= \exp\left(y \log \frac{\mu/n}{1-\mu/n} + n\log(1-\mu/n) + \log\binom{n}{y} \right) \\
&= \exp\left(y \log \frac{\mu}{n-\mu} + n\log(1-\mu/n) + \log\binom{n}{y} \right)
\end{aligned}$$

Exponential family

For the binomial distribution in the previous slide, we have $\phi = 1$, $\theta = \log \frac{\mu}{n-\mu}$, $b(\theta) = n \log(1 - \mu/n) = n \log(1 + e^{\theta})$ and $c(y, \phi) = \log \binom{n}{y}$

Similarly, for a Poisson(λ) distribution

$$f(y; \lambda) = e^{-\lambda}\lambda^{y}/y! = \exp(y \log(\lambda) - \lambda - \log y!)$$

We have $\phi = 1$, $\theta = \log \mu$, $b(\theta) = \mu = e^{\theta}$ and $c(y, \phi) = -\log y!$.

The parameter ϕ is generally referred to as a dispersion (i.e., “spread”) parameter.

For binomial and Poisson distributions, $\phi = 1$ while for a normal distribution $\phi = \sigma^2$

Question 1

- (b) Wald's test is based on the approximate normality of MLE estimate:

  $$\frac{\hat{\theta} - \theta_0}{se(\hat{\theta})} \dot{\sim} N(0, 1)$$

  We can use "pnorm()" function to find one-sided p-value. Compare this result with the p-value in summary output.

- (c) Drop in deviance test code is

  ```
  pchisq(DF_{deviance}, DF_{df.residual}, lowertail=FALSE)
  ```

- (e) we need to use "predict()" function.

Question 2

- (a) Two-sample t tests can be done using "t.test(A,B)". Make sure you include alternative = "greater" since we have a one-sided alternative hypothesis. Write a function if you can to save some typing.
- (b) Use drop-in-deviance test to formally test your smaller model is better than the null model.