Methods of Applied Statistics

Derek Li

Contents

1 Generalized Linear Model

1.1 Model

The GLM is

$$Y_i \sim G(\mu_i, \theta)$$
$$h(\mu_i) = X_i^T \beta$$

where G is the distribution of the response variable, μ_i is a location parameter for observation i, θ are additional parameters for the density of G, h is a link function, X_i are covariates for observation i, β is a vector of regression coefficients.

Example 1.1 (OLS)**.** G is a Normal distribution, θ is the variance parameter, denoted σ^2, h is the identity function, i.e.,

$$Y_i \sim \mathcal{N}(\mu_i, \sigma^2)$$
$$\mu_i = X_i^T \beta$$

Example 1.2 (Binomial/Logistic Regression)**.** G is a binomial distribution or a Bernoulli if $N_i = 1$, h is the logit link, $X_i^T \beta$ can be negative, μ_i is between 0 and 1, i.e.,

$$Y_i \sim \text{Binomial}(N_i, \mu_i)$$
$$\ln\left(\frac{\mu_i}{1-\mu_i}\right) = X_i^T \beta$$

In R, `glm` works like `lm` with a `family` argument. Binomial models in GLM require `y` to be a matrix with two columns `y` and `N-y`.

1.2 Inference: Parameter Estimation

$$\pi(Y_1, \cdots, Y_N; \beta, \theta) = \prod_{i=1}^{N} f_G(Y_i; \mu_i, \theta)$$
$$\ln L(\beta, \theta; y_1, \cdots, y_N) = \sum_{i=1}^{N} \ln f_G(y_i; \mu_i, \theta)$$

where Y_i are independently distributed, joint density π of random variables $(Y_1, \cdots, Y_N)$ is the product of the marginal densities f_G, likelihood function L given observed data $y_1, \cdots, y_N$ is a function of the parameters. The MLE is

$$(\widehat{\beta}, \widehat{\theta}) = \arg\max_{\beta,\theta} L(\beta, \theta; y_1, \cdots, y_N)$$

and the best parameters are those which are most likely to produce the observed data.

1.3 Inference

The information matrix is

$$I(\widehat{\beta}|Y) = \frac{\partial^2}{\partial\beta\partial\beta^T} - \ln L(\beta|Y)\bigg|_{\widehat{\beta}}$$

The MLEs are approximately Normal

$$\widehat{\beta} \sim \text{MVN}(\beta, I(\widehat{\beta}|Y)^{-1})$$

The standard errors are roots of diagonals of inverted information matrix.

```r
infMat = numDeriv::hessian(f=logLikelihood, x=fitcoef)
SE = sqrt(diag(solve(infMat)))
```