Exponential Families and Scoring

Fisher scoring is only useful when $J(\theta) = E\left[-d^2 L(\theta)\right]$ is easy to calculate, as in the Gamma example.

One class of distributions that satisfies this is the exponential family.

The density takes the form of

$$f\left(x\middle|\theta\right) = g(x)e^{\beta(\theta)+h(x)^T\gamma(\theta)}$$

This family of distributions includes the normal, binomial, poisson, gamma.

For example, with the gamma example

$$h(t)^T = \begin{bmatrix} t & \log t \end{bmatrix}$$

$$\gamma(\theta) = \begin{bmatrix} -\frac{1}{\lambda} & k \end{bmatrix}$$

$$\beta(\theta) = -k\log\lambda - \log\Gamma(k)$$

$$g(t) = \frac{1}{t}$$

For exponential families, the score function is

$$dL(\theta) = d\beta(\theta) + h(x)^T\, d\gamma(\theta)$$

and the Hessian is

$$d^2L(\theta) = d^2\beta(\theta) + h(x)^T\, d^2\gamma(\theta)$$

Note that if $d\gamma(\theta)$ is a linear function, then Newton-Raphson and Scoring are the same.

It can be shown that score and expected information can expressed in terms of $\mu(\theta) = E\left[h(X)\right]$ and $\Sigma(\theta) = \mathrm{Var}\left(h(X)\right)$.

It can be shown that $E\left[dL(\theta)\right] = 0$ since

$$E\left[dL(\theta)\right] = \int \frac{df\left(x|\theta\right)}{f\left(x|\theta\right)} f\left(x|\theta\right) d\nu(x)$$

$$= d\int f\left(x|\theta\right) d\nu(x)$$

and $\int f\left(x|\theta\right) d\nu(x) = 1$.

This statement can be restated as

$$d\beta(\theta) + \mu(\theta)^T\, d\gamma(\theta) = 0$$

Using this gives an alternate representation

$$dL(\theta) = \left[h(x) - \mu(\theta)\right]^T\, d\gamma(\theta)$$

i.e. the score is a sum of weighted residuals

From this we can immediately get the Information matrix as

$$J(\theta) = d\gamma(\theta)^T \Sigma(\theta) d\gamma(\theta)$$

Since it can be shown that $d\mu(\theta) = \Sigma(\theta) d\gamma(\theta)$, this implies the components of the scoring algorithm are

$$dL(\theta) = \left[h(x) - \mu(\theta)\right]^T \Sigma(\theta)^{-1} d\mu(\theta)$$

$$J(\theta) = d\mu(\theta)^T \Sigma(\theta)^{-1} d\mu(\theta)$$

Thus scoring for exponential families depends on the mean function (and its derivative) and the variance function of the sufficient statistic.

Note that in the case where $\Sigma(\theta)$ is not invertible, the above holds when a generalized inverse is used.

This is important for the multinomial as $\sum p_i = 1$ which implies a singular variance matrix.

Generalized Linear Models

One popular situation where the exponential family comes up is in Generalized Linear Models (GLIM) (McCullagh and Nelder, 1989)

In these models, include linear regression, logit and probit regression, and poisson regression.

In these models, the sufficient statistic $h(X) = X$ (sort of, I don't quite agree with the way Lange stated this)

The mean of X, $\mu(\theta)$ is postulated to have the form $q\left(z^T\theta\right)$, where q is a monotone function. The inverse is q is often referred to as the link function.

In this setting $d\mu(\theta) = q'\left(z^T\theta\right)z^T$, where z is the vector of covariates.

Then the score and expected information are

$$dL(\theta)^T = \sum_{i=1}^{m} \frac{x_i - \mu_i(\theta)}{\sigma_i^2} q'\left(z_i^T\theta\right) z_i$$

$$J(\theta) = \sum_{i=1}^{m} \frac{1}{\sigma_i^2} q'\left(z_i^T\theta\right)^2 z_i z_i^T$$

where $\sigma_i^2 = \text{Var}(X_i)$

Example: Logistic Regression

Low birth weight in Humans (Hosmer & Lemeshow, 1989).

Discussed in Venables and Ripley (page 222 3rd edition). Data available in dataframe `birthwt` in `MASS` library.

Response variable

$$y_i = \begin{cases} 0 & \text{birth weight} \geq 2.5\text{kg} \\ 1 & \text{birth weight} < 2.5\text{kg} \end{cases}$$

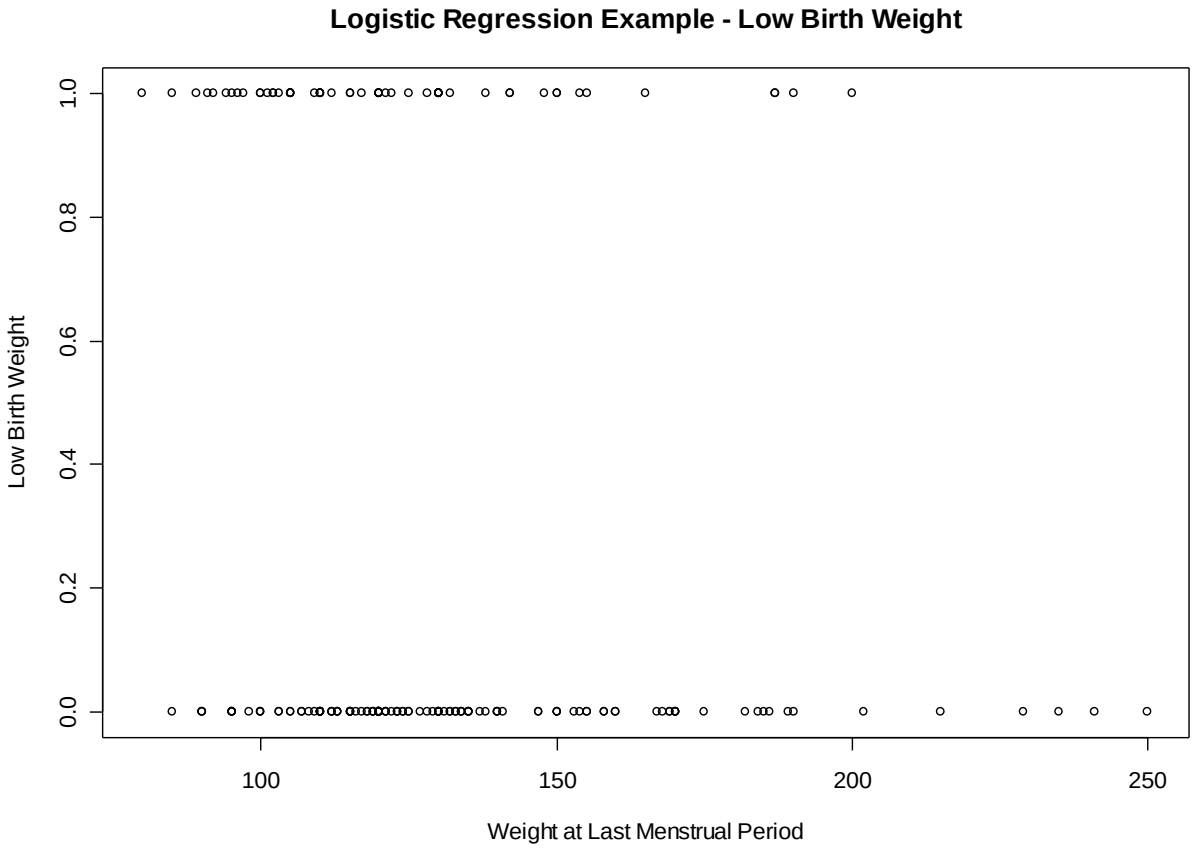
(`low` in dataframe)

Predictor variable

x_i = weight of mother (lbs) at last menstrual period (`lwt` in dataframe)

(In the actual data set, there are many more predictor variables available)

Want a model for the probability of having a low birth weight child, given the mother's weight.



The log likelihood for logistic regression is

$$L(\theta) = \sum_{i=1}^{m} \left\{ y_i \log \frac{e^{\theta_0 + \theta_1 x_i}}{1 + e^{\theta_0 + \theta_1 x_i}} + (1 - y_i) \log \frac{1}{1 + e^{\theta_0 + \theta_1 x_i}} \right\}$$

$$= \sum_{i=1}^{m} \left\{ y_i (\theta_0 + \theta_1 x_i) - \log\left(1 + \exp(\theta_0 + \theta_1 x_i)\right) \right\}$$

In general,

$$L(\theta) = \sum_{i=1}^{m} \left\{ y_i z_i \theta - \log\left(1 + \exp(z_i \theta)\right) \right\}$$

Link function:

$$q^{-1}(p) = \text{logit}(p) = \log \frac{p}{1-p}$$

$$q(t) = \frac{e^t}{1+e^t}$$

Mean function:

$$p(\theta) = q(\theta_0 + \theta_1 x) = \frac{e^{\theta_0 + \theta_1 x}}{1+e^{\theta_0 + \theta_1 x}}$$

Variance function:

$$\text{Var}(Y_i) = p(1-p)$$

$d\mu$ function:

$$q'(t) = \frac{e^t}{\left(1+e^t\right)^2} = \frac{q(t)}{1+e^t} = q(t)\left(1-q(t)\right)$$

These give the scoring components

$$dL(\theta) = \sum_{i=1}^{m} \begin{bmatrix} y_i - p_i \\ x_i (y_i - p_i) \end{bmatrix}$$

where $p_i = \frac{e^{\theta_0 + \theta_1 x_i}}{1+e^{\theta_0 + \theta_1 x_i}}$ and

$$J(\theta) = \sum_{i=1}^{m} \begin{bmatrix} 1 \\ x_i \end{bmatrix} \begin{bmatrix} 1 & x_i \end{bmatrix} \frac{1}{p_i(1-p_i)} \left(p_i(1-p_i) \right)^2$$

$$= \sum_{i=1}^{m} \begin{bmatrix} 1 \\ x_i \end{bmatrix} \begin{bmatrix} 1 & x_i \end{bmatrix} p_i(1-p_i)$$

Note that if you were to do Newton-Raphson, instead of scoring

$$d^2L(\theta) = -\sum_{i=1}^{m} \begin{bmatrix} 1 \\ x_i \end{bmatrix} \begin{bmatrix} 1 & x_i \end{bmatrix} p_i(1-p_i)$$

$$= -J(\theta)$$

For logistic regression, Newton-Raphson and Scoring are the same algorithms.

You can see this has to hold since $\sigma_i^2 = q'\left(z_i^T\theta\right)$.

Its actually an artifact of the piece of the log likelihood containing y_i being linear in the parameters (so y_i drops out of the 2nd partial derivatives)

$$L(\theta) = \sum_{i=1}^{m} \left\{ y_i(\theta_0 + \theta_1 x_i) - \log\left(1 + \exp(\theta_0 + \theta_1 x_i)\right) \right\}$$

This relationship will not hold for most GLIMs, such as Probit Regression, which is based on

$$p(\theta) = \Phi\left(z^T\theta\right)$$

$$q^{-1}(t) = \Phi^{-1}(t)$$

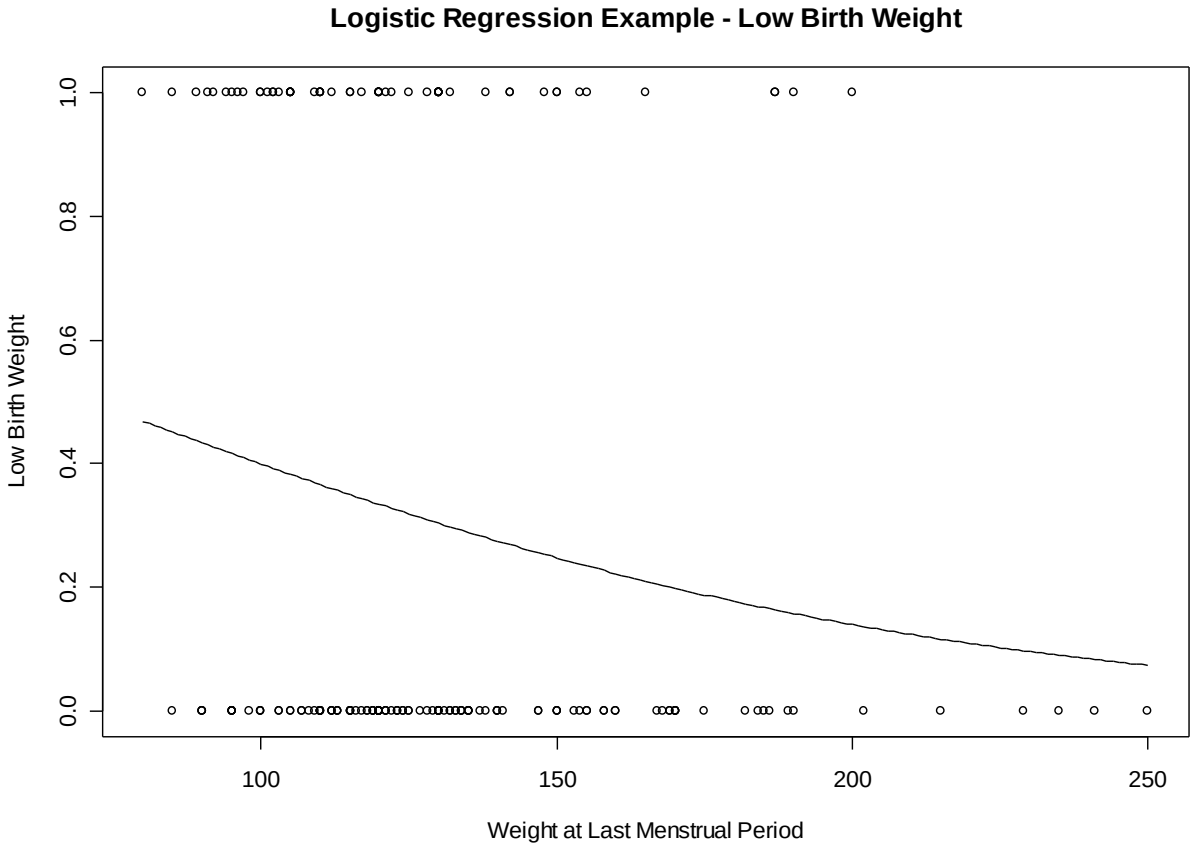
For the example, the iterations are

Iteration	θ_0	θ_1
0	0.8	0
1	0.5978497	-0.01204824
2	1.0083823	-0.01410487
3	0.9983194	-0.01405828
4	0.9983143	-0.01405826
5	0.9983143	-0.01405826

This was based on the convergence criterion

$$\max \frac{\left|\theta_i^n - \theta_i^{n-1}\right|}{\left|\theta_i^{n-1}\right| + 0.1} < 10^{-8}$$

The fitted curve showing the probability of a low birth weight is



The information and variance matrices are

$$J\left(\hat{\theta}\right) = \begin{bmatrix} 39.386 & 4908.917 \\ 4908.917 & 638101.268 \end{bmatrix}$$

$$\text{Var}\left(\hat{\theta}\right) = \begin{bmatrix} 0.616682 & -0.004744 \\ -0.004744 & 0.000038 \end{bmatrix}$$

The parameter estimates and their standard errors are

$$\hat{\theta} = [0.998314 \quad -0.014058]$$

$$\mathrm{SE}\left(\hat{\theta}\right) = [0.785290 \quad 0.006170]$$

For comparison, here is the output from R using `glm()`.

```
> summary(birthwt.glm)

Call:
glm(formula = low ~ lwt, family = binomial, data = birthwt)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.0951  -0.9022  -0.8018   1.3609   1.9821

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  0.99831    0.78529   1.271   0.2036
lwt         -0.01406    0.00617  -2.279   0.0227 *
---
Signif. codes:  0 `***' 0.001 `**' 0.01 `*' 0.05 `.' 0.1 ` ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 234.67  on 188  degrees of freedom
Residual deviance: 228.69  on 187  degrees of freedom
AIC: 232.69

Number of Fisher Scoring iterations: 4
```

Even though the log likelihood is unimodal (since $J(\theta)$ is positive definite everywhere), this doesn't guarantee convergence.

For example

Iteration	θ_0	θ_1
0	0.8	-0.3
1	-9.4734994	0.1032548
2	187.887804	-1.8589840
3	NaN	NaN

In this case, the poor starting values lead to jumping away from the optimum and instead head to the minimum out at ∞.

The Gauss-Newton Algorithm

An approach for models of the form

$$Y_i \sim N\left(\mu_i(\beta), \frac{\sigma^2}{w_i}\right)$$

where w_i are known constants and $\mu_i(\beta)$ will depend on covariates and is a nonlinear function of β.

The log likelihood is of the form

$$L(\theta) = -\frac{m}{2}\log\sigma^2 - \frac{1}{2\sigma^2}\sum_{i=1}^{m} w_i\left(y_i - \mu_i(\beta)\right)^2$$

The score and expected information are

$$dL(\theta) = \begin{bmatrix} \frac{1}{\sigma^2}\sum_{i=1}^{m} w_i\left(y_i - \mu_i(\beta)\right)d\mu_i(\beta) \\ -\frac{m}{2\sigma^2} + \frac{1}{2\sigma^4}\sum_{i=1}^{m} w_i\left(y_i - \mu_i(\beta)\right)^2 \end{bmatrix}$$

$$J(\theta) = \begin{bmatrix} \frac{1}{\sigma^2}\sum_{i=1}^{m} w_i d\mu_i(\beta)^T\, d\mu_i(\beta) & 0 \\ 0 & \frac{m}{2\sigma^4} \end{bmatrix}$$

Given the structure here, the scoring updates for β don't depend σ^2, so they can be done separately.

The update equation is

$$\beta_{n+1} = \beta_n + \left[\sum w_i d\mu_i(\beta_n)^T \, d\mu_i(\beta_n)\right]^{-1} \sum w_i (y_i - \mu_i(\beta_n)) d\mu_i(\beta_n)^T$$

Then

$$\hat{\sigma}^2 = \frac{1}{m}\sum_{i=1}^{m} w_i \left(y_i - \mu_i\left(\hat{\beta}\right)\right)^2$$

You can ignore the updates on σ^2 until the end.

Note that the piece of the log likelihood that depends on β is equivalent to a weighted least squares criteria

$$S(\beta) = \sum_{i=1}^{m} w_i \left(y_i - \mu_i(\beta)\right)^2$$

So minimizing $S(\beta)$ is equivalent to maximizing $L(\theta)$.

If you were to minimize $S(\beta)$ by Newton-Raphson, the Hessian is

$$d^2 S(\beta) = \sum_{i=1}^{m} w_i d\mu_i(\beta)^T \, d\mu_i(\beta) - \sum_{i=1}^{m} w_i \left(y_i - \mu_i(\beta)\right) d^2 \mu_i(\beta)$$

If we ignore the second term, the approximate Hessian is

$$d^2 S(\beta) \approx \sum_{i=1}^{m} w_i d\mu_i(\beta)^T \, d\mu_i(\beta)$$

which is what we get from scoring assuming normality.

So we can think of the scoring updates as Newton-Raphson for weighted nonlinear linear least squares with the Hessian replaced by positive definite matrix.

Iteratively Reweighted Least Squares (IRLS)

In some problems, the w_i are not known constants but functions of β, through the mean μ_i (e.g. $w_i(\beta) = f(\mu_i(\beta))$).

One case where this occurs is with GLIMs.

Notice that the score function for a GLIM can be written as

$$dL(\theta)^T = \sum_{i=1}^{m} \left(\sigma_i^2\right)^{-1} \left(x_i - \mu_i(\theta)\right) d\mu_i(\theta)$$

So another way of finding the parameter estimates is to use IRSL

1) Estimate $w_i = 1/\sigma_i^2(\theta_n)$

2) Minimize

$$S_n(\theta) = \sum_{i=1}^{m} w_i \left(y_i - \mu_i(\theta)\right)^2$$

3) Check for convergence. If not converged go back to 1.

I believe this is how many programs, including S-Plus and R actually fit GLIMs