Nonlinear Regression Models

$$E(Y|X) = g(X;\theta) + \epsilon,$$

g NOT linear in θ.

Fundamentally different (harder) than GLM.

Secs 8.1-8.6 in text.

Weight loss example:

X is days since start of weight reduction program

Y is weight on day X

Exponential decay model:

$$Y = \beta_0 + \beta_1 2^{-X/\lambda} + \epsilon,$$

all parameters have clear interpretations.

"Stormer Viscometer" (measures viscosity of a fluid) ex.:

Y is time taken for "inner cylinder to perform a fixed number of revolutions in response to an actuating weight."

X_1 is the viscosity

X_2 is the actuating weight

$$Y = \frac{\beta_1 X_1}{X_2 - \beta_2} + \epsilon$$

Calibration idea: run the experiment for a number of fluids of known viscosity, using a variety of weights. Use these data to fit the model and "learn" the parameters.

Then set to measure viscosities of other fluids, i.e., predict X_1 given Y and X_2 (interesting issues lurking here...).

Least-squares fitting (MLE assuming iid normal error terms)

Write as $Y_i = g^{(i)}(\theta) + \epsilon_i$

So

$$\hat{\theta} = \text{argmin}_\theta \sum_{i=1}^{n} \left\{ y_i - g^{(i)}(\theta) \right\}^2 .$$

No closed-form solution.

Newton-Raphson: fine (best) if happy to evaluate 1st and 2nd derivatives of $g^{(i)}(\theta)$.

Approach based on 1st derivative only?

Notation:

$$g_j^{(i)}(\theta) \equiv \partial g^{(i)}(\theta)/\partial\theta_j$$

Iterative scheme: let $\hat{\theta}^{(m)}$ be estimate after m iterations.

One-term Taylor-series approx. to linearize regression function:

$$\hat{\theta}^{(m+1)} = \text{argmin}_\theta \sum_{i=1}^{n} \left\{ y_i - g^{(i)}(\hat{\theta}^{(m)}) - \sum_{j=1}^{p} g_j^{(i)}(\hat{\theta}^{(m)})(\theta_j - \hat{\theta}_j^{(m)}) \right\}^2$$

i.e., get $\hat{\theta}^{(m+1)}$ as a linear least-squares fit, with a pseudo response vector and a pseudo design matrix.

General strategy for statistical computing: replace a hard problem with an iterative sequence of easy problems.

Convergence issues:

Need to provide $\hat{\theta}^{(0)}$ before we can start iterating.

As with virtually all optimazation problems, a 'bad' choice may produce non-convergence.

Possibilities for 'good' choices?

Scientific knowledge?

Somehow use the data to get a reasonable initial estimate. So called *self-starting* algorithm, perhaps based on fitting a related/simpler model?

For instance, think of weight-loss ex. Easy to automate initial estimation scheme for initial weight, total loss, 'half-life'.

Nitty-gritty implementation issues

nls() function in R / S. Lots of flexibility.

Besides starting value issue, need to decide how many derivatives to use: **zero, one, or two.**

In the case of zero, the algorithm will approximate first derivatives numerically. *This is not as easy/reliable as it sounds.*

In the case of one/two derivatives: user-supplied code to evaluate these derivatives or *symbolic manipulation* to do the work [deriv() function]? Latter sounds appealing...

Confidence intervals

General feeling that $\pm 1.96 SE$ intervals are less reliable in nonlinear problems. WHY?

ASIDE: SEs come from observed information matrix, i.e., from second derivative of log likelihood, which we may not have, or want to work at to get.

FIX: Let l_i be contribution of i-th datapoint to log-likelihood. May wish to replace

$$\hat{I} = -\sum_{i=1}^{n} l_i''(\hat{\theta}),$$

with

$$\hat{I} = \sum_{i=1}^{n} \left\{ l_i'(\hat{\theta}) \right\} \left\{ l_i'(\hat{\theta}) \right\}^T,$$

Alternative # 1: Invert a test.

Say $\theta = (\theta_1, \theta_2)$, want confidence region for θ_1.

Could test $H_0 : \theta_1 = \theta^*$ on the basis of

$$2\left\{\max_{\theta_1,\theta_2} l(\theta_1, \theta_2) - \max_{\theta_2} l(\theta^*, \theta_2)\right\} \quad \overset{approx}{\sim} \chi^2_{dim(\theta_1)}$$

if H_0 is true.

Thus, form 95% confidence region for θ_1 as all values of θ_* for which $H_0 : \theta_1 = \theta^*$ is NOT rejected by this test at significance level 0.05.

In contrast to SE-based (Wald) interval, such an interval

- may not be symmetric about $\hat{\theta}_1$

- will behave nicely under reparameterization.

Alternative #2: Bootstrap.