Nonlinear Regression Models

Temesgen H.

Dept. of Forest Engineering, Resources and Management

Oregon State University

Corvallis, Oregon

June 11, 2010

Outline

I. Background

II. Methods to find parameter estimates

III. Examples using R to fit nonlinear models

IV. Exercise

V. References

Why nonlinear models?

- Many processes (or attributes) are inherently nonlinear.
- Linear models do NOT describe asymptotic, or sigmoidal (s-shaped), or peaking functions (humped curves) well.
- Coefficients can be interpretable (meaningful)
- Possible extrapolation outside the range of the observed data

Methods to find parameter estimates:

1. Nonlinear least squares:

- objective is to find a set of coefficients that minimizes the sum of squared error (SSE, same as for OLS on linear models). Variances are then estimated separately.

Objective function is: $S(\hat{\boldsymbol{\theta}}) = \sum_{i=1}^{n} e_i^2 = \sum_{i=1}^{n} [y_i - f(\mathbf{x}_i, \theta)]^2$

2. Maximum likelihood:

- objective is to find the set of parameters (coefficients and variances) that maximizes the likelihood that you would get the sample data.

Finding Parameter Estimates for nonlinear models:

1. Choose a set of estimated coefficients (vector of starting values) to start your search (iteration=0) for the least squares solution.

Where should we get this starting set?

a) Previous fit using the same model on a similar dataset. Could be from published papers, or based on previous work you have done.

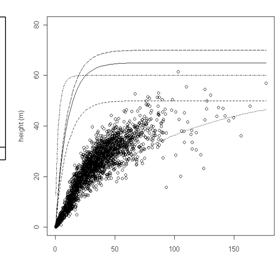
b) Fit a linear model that is very nearly the same as the nonlinear model, and use these coefficients as your starting set of parameters for the nonlinear search

c) Guess a logical set of starting parameters using physical or biological rules that have meaning for your model.

Where should we get this starting set (cont'd)?

d) Graphical exploration – use graphs to assess how different stating values mimic the data.

```
plot(ht2 ~ dbh, data=DF, xlab="dbh (cm)", ylab="height (m)",
ylim=c(0,80))
curve(expFct(x, b0=50, b1=-0.1, b2=1), add=TRUE, lty=2)
curve(expFct(x, b0=55, b1=-0.01, b2=0.9), add=TRUE, lty=3)
curve(expFct(x, b0=60, b1=-0.3, b2=0.6), add=TRUE, lty=4)
curve(expFct(x, b0=65, b1=-0.1, b2=1), add=TRUE, lty=1)
curve(expFct(x, b0=70, b1=-0.1, b2=1), add=TRUE, lty=5)
```



Where should we get this starting set (cont'd)?

e) Use a grid search:

- Choose several sets of possible coefficients
- Calculate the SSE for each set of coefficients.
- Select one of these sets that give the lowest SSE to use as starting parameters in your search.

E.g.: *grid.rat<-expand.grid(list(b0=seq(4,9, by=1), b1=c(-6), b2=seq(-1,-0.1, by=0.1)))*

Problems:

- Local optimum rather than a global optimum is obtained.
- The default number of iterations is reached, before you have a global optimum.

Possible solutions to obtain a global minimum:

- Try several sets of starting parameters to see if the same results occur.
- Use a search algorithm with both large and small stepsizes
- Try different algorithms and compare solutions. Should all achieve the same set of coefficients with the same minimum SSE. (e.g., Gauss-Newton, Marquardt's, etc).

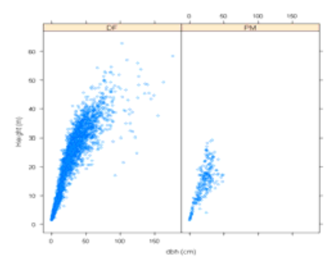
Assumptions of Nonlinear Least Squares

1) Error terms are normally distributed, and iid (as with OLS, independent, identically distributed (iid), means **errors independent and have the same variances**).

2) Assume first, second, third, etc. derivatives exist (i.e. the function is continuous).

Examples: Using R to Fit Nonlinear Regression

- Use a conditional scatter plot using the function xyplot() in the package lattice



Model fitting:

$$ht_i = 1.3 + \beta_0\left[1 - e^{\beta_1 dbh}\right]^{\beta_2} + \varepsilon_i$$

wher: β_0 asymptotic height, β_1 steepness parameter, and β_2 is curvature parameter that determines the rate of increase. β_0, β_1, and β_2 are species dependent coefficients, $\beta_0 > 0$, $\beta_1 < 0$, e is the Naperian constant

fit.CR<-nls(ht2~b0*((1-exp(b1*dbh))^b2), data=DF, start=list(b0=64, b1=-.01, b2=0.92),
+ trace=TRUE, nls.control(maxiter = 500))

Fitting C-R model using the nls() function

fit.CR<-nls(ht2~b0*((1-exp(b1*dbh))^b2), data=DF, start=list(b0=64, b1=-.01, b2=0.92),
+ trace=TRUE, nls.control(maxiter = 500))

The first argument, ht2~ b0*((1-exp(b1*dbh))^b2 is the model formula, where ~ is used to relate the response, ht2 (height-1.3), to the mean function, b0*((1-exp(b1*dbh))^b2, which is explicitly formulated on the right-hand side. Note that the predictor and the three parameters have to be specified explicitly, unlike the linear regression specification in lm().

The second argument (data) specifies the data frame containing the response and predictors. The argument *start* supplies the starting values for the parameters.

Comparing different models and fit statistics

To obtain the SSE, we use the deviance method.

deviance(fit.CR)

[1] 37637

The likelihood function, we use the logLik method

logLik(fit.CR)

'log Lik.' -6600 (df=4)

coef method to list the parameter estimates

b0	b1	b2
41.5701	-0.0324	1.1759

Model diagnostics:

A. Checking model form (mean structure) via plot of the fitted regression curve

Plot of the original data superimposed with the C-R fitted curve and examine the mean structure.

plot(ht2~dbh, data=DF, ylim=c(0,90), ylab="Height(m)", xlim=c(0,180), xlab="dbh (cm)")
dVal <-with(DF, seq(min(dbh), max(dbh), length.out=100))

lines(dVal, predict(fit.CR, newdata=data.frame(dbh=dVal)),col="red")
abline(h=1.3+coef(fit.CR)[1], lty=2, col="green")

Model diagnostics (cont'd):

B) Test for normality using the using the Shapiro-Wilk test:

> *shapiro.test(resid)*

C) Check assumptions of equal variance and normality of residuals (observed value - predicted value)

plot(fitted(fit.CR), residuals(fit.CR), xlab="Fitted Values", ylab="Residuals")

abline(a=0, b=0)

Please note heteroscedsacity - we will fix this problem towards the end of this or in the nonlinear mixed effect regression (NLME) session.

Model comparison:

We can use AIC to compare models; the smaller AIC the better. If more than 10 select one model over the other (Burnham and Anderson 2002).

> AIC(fit.CR)
[1] 13209

AIC(fit.rat)

> anova(fit.CR,fit.rat)
Analysis of Variance Table

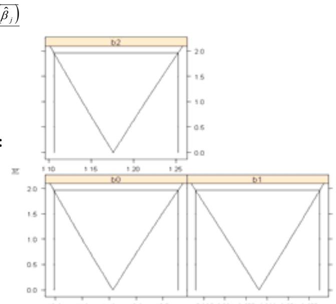
*Model 1: ht2 ~ b0 * ((1 - exp(b1 * dbh))^b2)*
Model 2: ht2 ~ exp(b0 + (b1/(dbh + b2)))
Res.Df Res.Sum Sq Df Sum Sq F value Pr(>F)
1 2350 37637
2 2350 37109 0 0
[1] 13175

For each parameter estimate, the profile t function is defined as (Ritz and Streibig 2008, p. 94)

$$\tau(\beta_j) = sign(\beta_j - \hat{\beta}_j)\frac{\sqrt{RSS(\beta_j) - RSS(\hat{\beta}_j)}}{s}$$

where s=residual standard error,

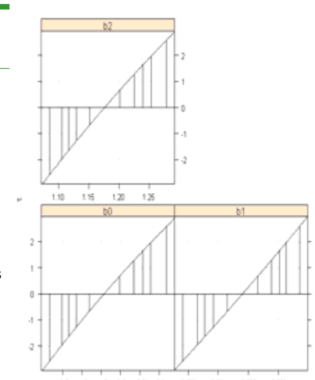
RSS= residual sums of squares.



Confidence intervals for the tau:

	2.5%	97.5%
b_0	40.4746	42.7817
b_1	-0.0354	-0.0295
b_2	1.1057	1.2526

Profile likelihood: indicates if linear approximation is perfect for each parameter or not.



Curvature of parameters: the linear approximation appears acceptable for b_1 and b_2, as there was not curvature for these parameter estimates.

Remedies for model violations:

A. Weighted least squares – weighted by a selected variable, etc.

B. Variance Modeling - explicitly models the variance; assumes errors are additive and normally distributed. For example, weights=varPower() specifies the variance model.

C. Transformations - uses Box-Cox regression

Acknowledgments

I thank:

Drs. Valerie LeMay, David Tait, Bianca Eskelson,
Andrew Robinson, and Derek Sattler