Nonlinear models
Paul Johnson 2010-09-18

1 Straight Lines are Great, but...

You like to think everything is linear:

$$y_i = b_0 + b_1 x_i + e_i$$

But you find a scatter plot that seems to indicate otherwise. What should you do?

2 Nonparametric regression: Draw a Curvy line

The title “nonparametric” model refers to approaches that “draw a curvy line” through a set of data.

Why “nonparametric”? Perhaps that’s a misleading title, because there are parameters hidden in most of these models. But... they are “nonparametric” in the sense that less emphasis is placed on fitting “the slope” or a particular “coefficient”. And these model’s don’t usually lead to “point estimates” and “t tests” in the same way that linear models do.

I have a separate handout on these things called NonparametricModels.

2.1 LOWESS (or LOESS): Locally weighted error sum of squares reduction.

Fit a reConsider each value of the input variable. Take the output for that value and try to predict it. Put most weight on values of the input that are close to the particular value being considered, put less weight on values of the input that are far away.

2.2 Splines: A “connect the dots” model.

Divide the input variable into regions, and create a predicted value function for each region. Draw “dots” at the predicted value at each “endpoint”, and connect the dots somehow.

2.2.1 A "linear spline" connects the dots with straight lines.

2.2.2 A "cubic spline" connects the dots with lines that are cubic functions, x^3.

2.2.3 A "natural spline" is a cubic spline with some restrictions on the end points.

2.2.4 A "thin plate spline" is a spline function in which the prediction depends on more than one variable.

3 Make up a complicated "parametric" formula.

Suppose "nature" uses a formula like this to "create" the data.

$$y_i = 14 \times x1_i \times e^{5.0+3.7\times log(x2_i)+u_i}$$

where u_i is an error term with $E(u_i) = 0$ and e represents "Euler's constant." You would hypthesize a formula of that same basic structure, but with letters representing parameters.

$$y_i = b_0 \times x1_i \times e^{b_1+b_2\times log(x2_i)+u_i}$$

Now, estimate parameters b_0, b_1, and b_2. If you had a really great estimator, you might come out close to the true values. At this point, properties like bias, consistency, efficiency, and asymptotic Normality come into play.

4 Intrinsically Linear models

Many nonlinear models can be algebraically re-arranged so that a simple re-coding of variables allows use of OLS.

4.1 Reciprocal

$$y_i = b_0 + b_1\frac{1}{x_i} + e_i \quad (1)$$

In the "old days", to estimate this model we would calculate a new variable:

$$recx_i = \frac{1}{x_i}$$

and then we would use ordinary least squares to estimate:

$$y_i = b_0 + b_1 recx_i + e_i \quad (2)$$

The estimates of the coefficients for $b_0$and $b_1$from this fit are applicable to the original model.

"Now days," some computer programs will let you write down a regression model using mathematical expressions, so you don't even have to do the job of creating the variables yourself.

[see R example of the reciprocal model at this point]

4.2 Double-log model

Suppose your theory were that

$$y_i = b_0 * x_i^{b_1} * e_i \tag{3}$$

The coefficients have a specific interpretation, so this might be worthwhile. Several political science articles use this.

Note if you log both sides, you end up with a model that is intrinsically linear:

$$ln(y_i) = ln(b_0) + b_1 * ln(x_i) + ln(e_i) \tag{4}$$

Supposing that

$$E[ln(e_i)] = 0 \tag{5}$$

and

$$E[((ln(e_i))^2] = \sigma^2 \tag{6}$$

then all is well. In fact, we might as well rename the error term,

$$u_i = ln(e_i) \tag{7}$$

Just create new variables:

$$ylog_i = ln(y_i)\ and\ xlog_i = ln(x_i) \tag{8}$$

and then fit this with OLS

$$ylog_i = c_o + c_1 xlog_i + u_i \tag{9}$$

The estimate of $c_0$is an estimate of $ln(b_0)$,so you can convert $\hat{c_0}$to $\hat{b_0}$with

$$\hat{b_0} = exp(\hat{c_0}) \tag{10}$$

and the estimate of $b_1$is the same as the estimate of c_1.

4.3 Log on the left

$$y_i = exp(b_0 + b_1 x_i + e_i) \tag{11}$$

That's nonlinear, but if you log both sides, it gets pretty darned linear.

$$ln(y_i) = b_0 + b_1 x_i + e_i \tag{12}$$

4.4 Log on the right

$$y_i = b_0 + b_1 * log(x_i) + e_i \tag{13}$$

4.5 Polynomial

$$y_i = b_0 + b_1 * x_i + b_2 * x_i^2 + e_i \tag{14}$$

5 Theory has to guide you on choosing a formula.

The data will seldom give you a good reason to pick one model over another.

If the left hand side of the expression is the same, then R^2can serve as a guide for picking the best fitting model, but if I had a good theoretical justification to pick one model over another, I'd do that.

6 Interpretation of results is the most important part.

Consider the effort you make to interpret an OLS model. "Each unit increase in x causes a $\hat{b_1}$ increase in the expected value of y.

You need to make a similar effort to interpret a nonlinear model, remembering that each one has unique mathematical properties.

Usually these are things to look for:

1. Can you understand the slope of the line representing the expected value?
2. Does the function have a maximum value that is substantively important?
3. Are there any "special" values of the parameters that you need to watch out for and give special interpretation.

7 Nonlinear models that won't yield to OLS

Sometimes you can find a nonlinear model that can't be transformed to estimate with OLS. There are many other approaches.

7.1 Maximum Likelihood

One approach is maximum likelihood. I have a separate handout on that. Currently, it is distributed as "mlNotes1".

7.2 Nonlinear least squares

Another approach is nonlinear least squares (NLS). The nonlinear least squares approach used to seem like a "magical, mysterious unknown" to me because we did not have software to do it. Now there is nonlinear least squares in just about every stat package. Basically, you write down a formula, assume the error is additive, and go:

$$y_i = f(X_i, b) + e_i \tag{15}$$

Typically the first-level assumption is that the error is Normal, meaning $y_i$is Normal, and you fit by making the sum of squared errors the smallest. Minimize this:

$$SS(\hat{b}) = \sum_{i=1}^{N} [y_i - f(X_i, \hat{b})]^2 \tag{16}$$

Now, if you suppose the error is not Normal, then y will not be Normal. That puts you off into the territory of the "generalized linear regression model". A generalized linear model supposes that the "predictive part" $f(X_i,b)$ is just a linear formula and the problem is to use that straight line to calculate the expected value of some nonNormal variable.

If you hypothesize that the "predictive part" is nonlinear, then you enter the province of the "generalized nonlinear regression model." Those things exist!

I wish I had exhaustive knowledge of NLS, but I don't, so I stop here and say the following.

I don't believe the NLS parameter estimates are unbiased, but I believe they are efficient and consistent. I also believe they are, in the limit (for large sample sizes) equivalent to maximum likelihood estimates. Frequently, when a problem is too complicated for ML, an author will derive a least squares approach and then rely on the claim that it is equivalent to ML for large samples.

8 Generalized Additive Models (GAM)

Start with the idea of fitting a predictor $x1_i$ with a loess or natural spline.

$$\hat{y}_i = f(x1_i) \tag{17}$$

If you want to, you could assume another variable $x2_i$ has an effect that is totally unrelated to $x1_i$. If so, you could make a loess or spline for that variable.

$$\hat{y}_i = f(x1_i) + g(x2_i) \tag{18}$$

This is a **generalized additive model.** It is additive–linear–because the separate variables have separate influences.

You have a **semi-parametric model** if you fit a loess or spline for $x1_i$ but then want to estimate a linear model for $x2_i$.

$$\hat{y}_i = f(x1_i) + b_2 x2_i \tag{19}$$

9 Very Clever Parametization: Box-Cox

Box & Cox proposed a way to transform variables that is rather general. The transformation depends on a parameter λ, which can take on values in a continuum.

$$y^{(\lambda)} = \begin{cases} \frac{y^{\lambda}-1}{\lambda} & if\ \lambda \neq 0 \\ log(y) & if\ \lambda = 0 \end{cases}$$

Put in some sample values for λ and you will start to see what this transformation does. If $\lambda = 0$, then the transformation is simply the natural log. If $\lambda = 1$,then the transformation simply subtracts one from the variable.

What should λ be? If you have a very good knowledge of your data, you can just set the "right" value. Also, it can be estimated in an iterative 2 step process. Hypothesize a value for λ, and then use that value to transform the data and fit a model. Then hypothesize another value for λ, re-fit the model, and if it fits better, then adopt the new value of λ.

I have a separate handout on these models called BoxCoxRegression.