STAT 705 Generalized additive models with non-normal responses

Adapted from Timothy Hanson

Department of Statistics, University of South Carolina

Stat 705: Data Analysis II

Review from STAT 704

Consider a linear regression problem:

$$Y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \epsilon_i,$$

where $e_1, \ldots, e_n \overset{iid}{\sim} N(0, \sigma^2)$.

- Diagnostics (residual plots, added variable plots) might indicate poor fit of the basic model above.
- Remedial measures might include transforming the response, transforming one or both predictors, or both.
- One also might consider adding quadratic terms and/or an interaction term.

When considering a transformation of one predictor, an added variable plot can suggest a transformation (e.g. $\log(x), 1/x$) that might work *if the other predictor is "correctly" specified*.

In general, a transformation is given by a function $x^* = g(x)$. Say we decide that x_{i1} should be log-transformed and the reciprocal of x_{i2} should be used. Then the resulting model is

$$\begin{aligned} Y_i &= \beta_0 + \beta_1 \log(x_{i1}) + \beta_2/x_{i2} + \epsilon_i \\ &= \beta_0 + g_{\beta_1}(x_{i1}) + g_{\beta_2}(x_{i2}) + \epsilon_i, \end{aligned}$$

where $g_{\beta_1}(x)$ and $g_{\beta_2}(x)$ are two functions specified by β_1 and β_2.

Here we are specifying forms for $g_1(x|\beta_1)$ and $g_2(x|\beta_2)$ based on exploratory data analysis, but we could from the outset specify *models* for $g_1(x|\boldsymbol{\theta}_1)$ and $g_2(x|\boldsymbol{\theta}_2)$ that are rich enough to capture interesting and predictively useful aspects of how the predictors affect the response and *estimate these functions from the data*.

One example of this is through a basis expansion; for the jth predictor the transformation is:

$$g_j(x) = \sum_{k=1}^{K_j} \theta_{jk}\psi_{jk}(x),$$

where $\{\psi_{jk}(\cdot)\}_{k=1}^{K_j}$ are B-spline basis functions, sine/cosine functions, etc. This is not the approach taken in SAS PROC GAM. PROC GAM makes use of *cubic smoothing splines*.

This is an example of "nonparametric regression," which ironically connotes the inclusion of *lots* of parameters rather than fewer.

For simple regression data $\{(x_i, y_i)\}_{i=1}^n$, a cubic spline smoother $g(x)$ minimizes

$$\sum_{i=1}^{n}(y_i - g(x_i))^2 + \lambda \int_{-\infty}^{\infty} g''(x)^2 dx.$$

Good fit is achieved by minimizing the sum of squares $\sum_{i=1}^n (y_i - g(x_i))^2$. The $\int_{-\infty}^{\infty} g''(x)^2 dx$ term measures how wiggly $g(x)$ is and $\lambda \geq 0$ is how much we will penalize $g(x)$ for being wiggly.

So the spline trades off between goodness of fit and wiggliness.

Although not obvious, the solution to this minimization is a cubic spline: a piecewise cubic polynomial with the pieces joined at the unique x_i values.

Cubic spline basis

Cubic splines are piecewise cubic polynomial functions with *knots* $\{\tau_k\}$, where the knots are typically defined as unique values of X. Assume each of the n observed values of X are unique in the following discussion.

- The basis consists of truncated cubic polynomials "centered" at each of the knots, augmented by lower order polynomials.

$$\phi_1(t) = 1,\ \phi_2(t) = t,\ \phi_3(t) = t^2,\ \phi_4(t) = t^3,$$
$$\phi_{k+4}(t) = (t - \tau_k)_+^3,\ k = 1, \ldots, n$$

Cubic spline basis

- The piecewise cubic polynomial components are continuous and have continuous first and second derivatives.
- The cubic polynomial "joins" are smooth to the eye.
- *Natural* cubic spline bases are linear beyond the knots; PROC GAM estimates a natural cubic spline basis.
- The truncated power basis can result in a nearly singular X matrix; B-spline bases are more numerically stable.

Hastie and Tibshirani (1986, 1990) point out that the meaning of λ depends on the units x_i is measured in, but that λ can be picked to yield an "effective degrees of freedom" *df* or an "effective number of parameters" being used in $g(x)$. Then the complexity of $g(x)$ is equivalent to $(df - 1)$-degree polynomial, but with the coefficients "spread out", yielding a more flexible function that fits data better.

Alternatively, λ can be picked through cross validation, by minimizing

$$CV(\lambda) = \sum_{i=1}^{n} (y_i - g_{\lambda}^{-i}(x_i))^2.$$

Both options are available in SAS.

We have $\{(\mathbf{x}_i, y_i)\}_{i=1}^n$, where $y_1, \ldots, y_n$ are normal, Bernoulli or Poisson. The generalized additive model (GAM) is given by

$$h\{E(Y_i)\} = \beta_0 + g_1(x_{i1}) + \ldots + g_p(x_{ip}).$$

Each of $g_1(x), \ldots, g_p(x)$ are modeled via cubic smoothing splines, each with their own smoothness parameters $\lambda_1, \ldots, \lambda_p$ either specified as $df_1, \ldots, df_p$ or estimated through cross-validation. The model is fit through *back-fitting* and *local scoring*. See Hastie and Tibshirani (1990) or the SAS documentation for details.

PROC GAM in SAS

SAS actually fits $g_j(x_j) = \beta_j x_j + \tilde{g}_j(x_j)$, where $\tilde{g}_j(x_j)$ integrates to zero over the range of x_j. Thus one can test $H_0 : \tilde{g}_j(\cdot) = 0$, i.e. the usual linear predictor is sufficient for x_j.

In analyzing whether to reject $H_o : \tilde{g}_j(\cdot) = 0$, we would like to look at the plot of $g_j(x_j) = \beta_j x_j + \tilde{g}_j(x)$ versus x spanning the range of $x_{1j}, \ldots, x_{nj}$. This is provided in the R package GAM (also in DPpackage, a Bayesian version), but not in SAS (SAS plots only $\tilde{g}_j(x)$ versus x, as we will see).

Satellite counts Y_i

Let's fit a GAM to the horseshoe crab mating data:

```
proc gam plots(unpack)=components(clm) data=crabs;
 class spine color;
 model satellite=param(color) spline(width)  / dist=poisson;
run;
```

This fits the model

$$Y_i \sim \text{Pois}(\mu_i),$$

$$\begin{aligned}\log(\mu_i) =& \beta_0 + \beta_1 I\{c_i = 1\} + \beta_2 I\{c_i = 2\} + \beta_3 I\{c_i = 3\} \\ &+ \beta_4 \times \text{width}_i + \tilde{g}_4(\text{width}_i).\end{aligned}$$

SAS parameter estimates

Regression Model Analysis
Parameter Estimates

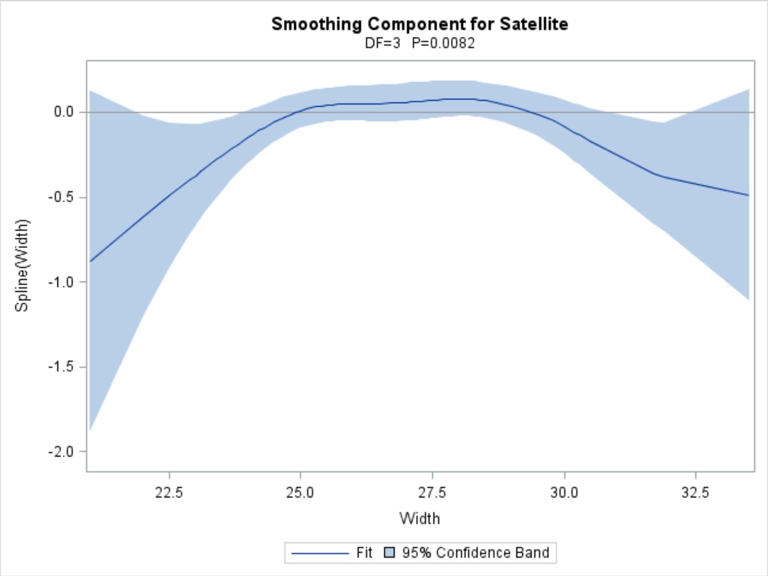
Parameter	Parameter Estimate	Standard Error	t Value	Pr > |t|
Intercept	-3.09884	0.60839	-5.09	<.0001
color 1	0.39679	0.20871	1.90	0.0590
color 2	0.23341	0.16249	1.44	0.1528
color 3	-0.00422	0.18041	-0.02	0.9814
color 4	0	.	.	.
Linear(width)	0.15032	0.02270	6.62	<.0001

Smoothing Model Analysis
Analysis of Deviance

Source	DF	Sum of Squares	Chi-Square	Pr > ChiSq
Spline(width)	3.00000	11.777666	11.7777	0.0082

The Analysis of Deviance table gives a χ^2-test from comparing the deviance between the full model and the model with this variable dropped: here the model with color (categorical) *plus only a linear effect in weight*. We see that width is significantly nonlinear at the 5% level. The default $df = 3$ corresponds to a smoothing spline with the complexity of a cubic polynomial.

The following plot has the estimated smoothing spline function with the linear effect subtracted out. The plot includes a 95% confidence band for the whole curve. We visually inspect which portions of this band do not include zero to get an idea where significant nonlinearity occurs. This plot can suggest simpler transformations of predictor variables than use of the full-blown smoothing spline: here maybe a quadratic?


Smoothing Component for Satellite
DF=3 P=0.0082
Spline(Width)
0.0
-0.5
-1.0
-1.5
-2.0
22.5
25.0
27.5
30.0
32.5
Width
Fit
95% Confidence Band

The band shows a pronounced deviation from linearity for weight. The plot spans the range of weight values in the data set and becomes highly variable at the ends. Do you think extrapolation is a good idea using GAMs?

Note: You *can* get predicted values out of SAS with CIs. Just stick to representative values.

GAM in R

The package `gam` was written by Trevor Hastie (one of the inventors of GAM) and (in your instrucgtor's opinion) is easier to use and provides better output than SAS PROC GAM.

Just as in PROC GAM, you provide the function `gam` a list of transformed and untransformed predictors. Note that it does not make sense to consider a transformation of a categorical predictor.

The `gam` function provies plots of the ful transformation $g_j(\cdot)$, not just the "wiggly" part $\tilde{g}_j(\cdot)$.

O-ring data

- Motivation: explosion of USA Space Shuttle Challenger on January 28, 1986.
- Rogers commission concluded that the Challenger accident was caused by gas leak through the six o-ring joints of the shuttle.
- Dalal, Fowlkes & Hoadley (1989) looked at the number of distressed o-rings (among 6) versus launch temperature (Temperature) and pressure (Pressure) for 23 previous shuttle flights, launched at temperatures between 53 °F and 81 °F.

O-ring variables

- ThermalDistress: a numeric vector indicating whether the o-ring experienced thermal distress.
- Temperature: a numeric vector giving the launch temperature ($^\circ$F).
- Pressure: a numeric vector giving the leak-check pressure (psi).
- Flight: a numeric vector giving the temporal order of flight.

Dalal, S.R., Fowlkes, E.B., and Hoadley, B. (1989). Risk analysis of space shuttle: Pre-Challenger prediction of failure. *Journal of the American Statistical Association*, 84, 945-957.

Analysis in R

```
library(DPpackage); library(gam)
data(orings)
?orings
plot(orings)  # note that pressure only has three values
fit-gam(ThermalDistress~s(Temperature)+Pressure+s(Flight),
family=binomial(link=logit),data=orings)
par(mfrow-c(2,2))
plot(fit,se=TRUE)
summary(fit)
```

This fits the model

$$logit(\pi_j) = \beta_0 + \beta_1 T_j + \beta_2 P_j + \beta_3 F_j + \tilde{g}_1(T_j) + \tilde{g}_3(F_j).$$