Lecture 24: Generalized Additive Models
Stat 704: Data Analysis I, Fall 2010

Tim Hanson, Ph.D.

University of South Carolina
Generalized additive models

Consider a linear regression problem:

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

where \( \epsilon_1, \ldots, \epsilon_n \) 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.
* Note: we only consider transforming continuous predictors!
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

\[
Y_i = \beta_0 + \beta_1 \log(x_{i1}) + \frac{\beta_2}{x_{i2}} + \epsilon_i
\]

\[
= \beta_0 + g_{\beta_1}(x_{i1}) + g_{\beta_2}(x_{i2}) + \epsilon_i,
\]

where \(g_{\beta_1}(x)\) and \(g_{\beta_2}(x)\) are two functions specified by \(\beta_1\) and \(\beta_2\).
Generalized additive models

Additive predictors

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 \( g_1(x|\theta_1) \) and \( g_2(x|\theta_2) \) 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 an basis expansion; for the \( j \)th 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, or sines/cosines, etc. This approach has gained more favor from Bayesians, but 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.
Hastie and Tibshirani (1986, 1990) point out that the meaning of $\lambda$ depends on the units $x_i$ is measured in, but that $\lambda$ 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” more yielding a more flexible function that fits data better.

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

$$h\{E(Y_i)\} = \beta_0 + g_1(x_{i1}) + \cdots + g_k(x_{ik}),$$

for $p$ predictor variables. $Y_i$ is a member of an exponential family such as binomial, Poisson, normal, etc. $h$ is a link function.

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 “backfitting.” See Hastie and Tibshirani (1990) or the SAS documentation for details.
Satellite counts $Y_i$

Let’s fit a GAM to the crab mating data:

```plaintext
ods pdf; ods graphics on;
proc gam plots(unpack)=components(clm) data=crabs;
  class spine color;
  model satell=param(color) spline(weight) / dist=poisson;
run;
ods graphics off; ods pdf close;
run;
```

This fits the model

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

$$\log(\mu_i) = \beta_0 + \beta_1 I\{c_i = 1\} + \beta_2 I\{c_i = 2\} + \beta_3 I\{c_i = 3\} + \beta_4 wt_i + g_4(wt_i).$$
SAS output

Output:

The GAM Procedure
Dependent Variable: satell
Regression Model Component(s): color
Smoothing Model Component(s): spline(weight)

Summary of Input Data Set

<table>
<thead>
<tr>
<th>Number of Observations</th>
<th>173</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of Missing Observations</td>
<td>0</td>
</tr>
<tr>
<td>Distribution</td>
<td>Poisson</td>
</tr>
<tr>
<td>Link Function</td>
<td>Log</td>
</tr>
</tbody>
</table>

Class Level Information

<table>
<thead>
<tr>
<th>Class</th>
<th>Levels</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>color</td>
<td>4</td>
<td>1, 2, 3, 4</td>
</tr>
</tbody>
</table>

Iteration Summary and Fit Statistics

<table>
<thead>
<tr>
<th>Number of local scoring iterations</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>Local scoring convergence criterion</td>
<td>3.01103E-11</td>
</tr>
<tr>
<td>Final Number of Backfitting Iterations</td>
<td>1</td>
</tr>
<tr>
<td>Final Backfitting Criterion</td>
<td>2.286359E-10</td>
</tr>
<tr>
<td>The Deviance of the Final Estimate</td>
<td>532.81821791</td>
</tr>
</tbody>
</table>
SAS parameter estimates

Regression Model Analysis
Parameter Estimates

| Parameter     | Estimate | Standard Error | t Value | Pr > |t| |
|---------------|----------|----------------|---------|------|---|
| Intercept     | -0.50255 | 0.23759        | -2.12   | 0.0359 |
| color 1       | 0.36148  | 0.20850        | 1.73    | 0.0848 |
| color 2       | 0.21891  | 0.16261        | 1.35    | 0.1801 |
| color 3       | -0.01158 | 0.18063        | -0.06   | 0.9490 |
| color 4       | 0        | .              | .       | .    |
| Linear(weight) | 0.56218  | 0.07894        | 7.12    | <.0001 |

Smoothing Model Analysis
Analysis of Deviance

<table>
<thead>
<tr>
<th>Source</th>
<th>DF</th>
<th>Sum of Squares</th>
<th>Chi-Square</th>
<th>Pr &gt; ChiSq</th>
</tr>
</thead>
<tbody>
<tr>
<td>Spline(weight)</td>
<td>3</td>
<td>18.986722</td>
<td>18.9867</td>
<td>0.0003</td>
</tr>
</tbody>
</table>
The Analysis of Deviance table gives a $\chi^2$-test from comparing the deviance between the full model and the model with this variable dropped – here the intercept model plus a linear effect in temperature. We see that weight 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 where this band does not include zero to get an idea of 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?
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.
PROC GAM handles Poisson, Bernoulli, normal, and gamma data. If you only have normal data, PROC TRANSREG will fit a very general transformation model, for example

\[ h(Y_i) = \beta_0 + g_1(x_{i1}) + g_2(x_{i2}) + \epsilon_i, \]

and provide estimates of \( h(\cdot), g_1(\cdot), \) and \( g_2(\cdot). \)

\( h(\cdot) \) can simply be the Box-Cox family, indexed by \( \lambda \), or a very general spline function.
* Consider time-to-failure in minutes of $n = 50$ electrical components.
* Each component was manufactured using a ratio of two types of materials; this ratio was fixed at 0.1, 0.2, 0.3, 0.4, and 0.5.
* Ten components were observed to fail at each of these manufacturing ratios in a designed experiment.
* It is of interest to model the failure-time as a function of the ratio, to determine if a significant relationship exists, and if so to describe the relationship simply.
## SAS code: data & plot

```sas
data elec;
input ratio time @@;
datalines;
0.5 34.9 0.5 9.3 0.5 6.0 0.5 3.4 0.5 14.9 0.5 9.0 0.5 19.9 0.5 2.3 0.5 4.1 0.5 25.0
0.4 16.9 0.4 11.3 0.4 25.4 0.4 10.7 0.4 24.1
0.4 3.7 0.4 7.2 0.4 18.9 0.4 2.2 0.4 8.4
0.3 54.7 0.3 13.4 0.3 29.3 0.3 28.9 0.3 21.1
0.3 35.5 0.3 15.0 0.3 4.6 0.3 15.1 0.3 8.7
0.2 9.3 0.2 37.6 0.2 21.0 0.2 143.5 0.2 21.8
0.2 50.5 0.2 40.4 0.2 63.1 0.2 41.1 0.2 16.5
0.1 373.0 0.1 584.0 0.1 1080.1 0.1 300.8 0.1 130.8
0.1 280.2 0.1 679.2 0.1 501.6 0.1 1134.3 0.1 562.6
;
ods pdf; ods graphics on;
proc sgscatter; plot time*ratio; run;
ods graphics off; ods pdf close;
```
SAS code: fit \( h(Y_i) = \beta_0 + g_1(x_{i1}) + \epsilon_i \)

ods pdf; ods graphics on;
proc transreg data=elec solve ss2 plots=(transformation obp residuals);
  model spline(time) = spline(ratio); run;
ods graphics off; ods pdf close;
What to do?

* The “best” fitted transformations look like log or square roots for both time and ratio.
* The log is also suggested by Box-Cox for time (not shown). Code:
  
  `model boxcox(time) = spline(ratio)`
* Refit the model with these simple functions:
* `model log(time) = log(ratio)`
Generalized additive models

T. Hanson (USC)
* Better, but not perfect.
* What if we transform $Y_i$ first, then look at a simple scatterplot of the data?
* Here is plot of log(time) versus ratio...what transformation would you suggest for ratio?