

# Generalized Additive Model

by

Huimin Liu

Department of Mathematics and Statistics  
University of Minnesota Duluth, Duluth, MN 55812

December 2008

## Table of Contents

<b>Abstract</b> .....	2
<b>Chapter 1 Introduction</b>	
1.1 Regression Review .....	3
1.2 Introduce to Generalized Additive Model (GAM) .....	4
<b>Chapter 2 Smoothing</b>	
2.1 What it a Smoother.....	5
2.2 Cubic Smoothing Splines.....	5
2.3 Automatic selection of smoothing parameter $\lambda$ .....	8
2.3.1 From MSE to PSE.....	8
2.3.2 Cross Validation.....	10
2.3.3 Generalized Cross Validation.....	12
2.3.4 Degrees of Freedom of a Smoother.....	15
<b>Chapter 3 Generalized Additive Model (GAM)</b>	
3.1 Generalized Additive Model (GAM).....	17
3.2 Fitting Generalized Additive Model.....	17
3.3 The Logistic Generalized Additive Model.....	21
<b>Chapter 4 Examples</b>	
4.1 Example 1: Comparing GAM with GLM.....	25
4.2 Example 2: Fitting logistic GAM model using GAM procedure.....	28
<b>Chapter 5</b>	
<b>Discussion</b> .....	35
<b>References</b> .....	36
<b>Appendix</b> .....	37

## Abstract

Generalized linear model assumes a linear relationship between the mean of the

dependent variable and the unknown parameter  $\beta_1, \dots, \beta_p$ .  $\mu_{y|x} = E(y|x) = \beta_0 + \sum_{j=1}^p \beta_j X_j$  or

$g(\mu_{y|x}) = \beta_0 + \sum_{j=1}^p \beta_j X_j$ , where  $X_1, \dots, X_p$  are independent variables. In this paper, we

study the class of Generalized Additive models [Hastie and Tibshirani. (1990)], which

replace the linear form  $\sum_{j=1}^p \beta_j X_j$  by a sum of smooth functions  $\sum_{j=1}^p f_j(x_j)$ . This non-

parametric function  $f_j(x_j)$  can be estimated in a flexible manner using cubic spline

smoother, in an iterative method called the back fitting algorithm. Generalized Additive

models are suitable for exploring the data set and visualizing the relationship between the

dependant variable and the independent variables. This paper illustrates the technique by

comparing the PROC GAM procedure with the PROC GLM procedure by using SAS and

uses PROC GAM to model data from binary distribution as well.

## Keywords:

Generalized Additive Models (GAM), cubic spline smoother, non-parametric, back-fitting

## Chapter 1 Introduction

### 1.1 Regression Review

One of the most popular and useful tools in data analysis is the linear regression model. It is a statistical technique used for modeling and analysis of numerical data consisting of values of a dependent variable and of one or more independent variables.

Let  $Y$  be a dependent (response) variable, and  $X_1, \dots, X_p$  be  $p$  independent (predictor or regressor) variables. Our goal is to describe the dependence of the mean of  $Y$  as a function of  $X_1, \dots, X_p$ . For this purpose, we assume that the mean of  $Y$  is a linear function of  $X_1, \dots, X_p$ ,

$$\begin{aligned}\mu_{y|x} &= E(Y | X_1, \dots, X_p) = f(X_1, \dots, X_p) = \beta_0 + \beta_1 X_1 + \dots + \beta_p X_p \\ &= \beta_0 + \sum_{j=1}^p \beta_j X_j\end{aligned}$$

$$\text{or } g(\mu_{y|x}) = \beta_0 + \sum_{j=1}^p \beta_j X_j \text{ for some function } g. \quad (1.1.1)$$

Given a sample of values for  $Y$  and  $X$ , where  $X = (X_1, \dots, X_p)$ , the estimates of

$\beta_0, \beta_1, \dots, \beta_n$  are often obtained by the least squares method. It is achieved by fitting a

linear model which minimizes  $\sum_{j=1}^p (y_j - f_j(\hat{x}_j))^2$ , where  $f(\hat{x}_j) = \hat{\beta}_j x_j$ .

## 1.2 Introduction to the Generalized Additive Model (GAM)

The Generalized additive model replaces  $\sum_{j=1}^p \beta_j X_j$  with  $\sum_{j=1}^p f_j(x_j)$  where  $f_j$  is an unspecified ('non-parametric') function. It can be in a non-linear form:

$$\begin{aligned} E(Y | X_1, \dots, X_p) &= f(X_1, \dots, X_p) = f_0 + f_1(X_1) + \dots + f_p(X_p) \\ &= f_0 + \sum_{j=1}^p f_j(X_j) \quad j = 1, \dots, p \end{aligned} \quad (1.2.1)$$

This function  $\hat{f}_j(x_j)$  is estimated in a flexible manner using cubic spline smoother.

## Chapter 2 Smoothing

### 2.1 What is a Smoother?

A smoother is a tool for summarizing the trend of a dependent variable  $Y$  as a function of one or more independent variables  $X_1, \dots, X_p$ . It produces an estimate of the trend that is less variable than  $Y$  itself; hence named *smoother*. We call the estimate produced by a smoother a *smooth*.

Smoother is very useful in statistical analysis. First, we can pick out the trend from the plot easily. Second, it estimates the dependence of the mean of  $Y$  on the predictors.

The most important property of smoother is its non-parametric nature. So the smooth function is also known as non-parametric function. It doesn't assume a rigid form for the dependence of  $Y$  on  $X_1, \dots, X_p$ . This is the biggest difference from Generalized Linear Model (GLM). It allows an 'approximation' with sum of functions, (these functions have separated input variables), not just with one unknown function only. That's why it is the building block of the generalized additive model algorithm.

### 2.2 Cubic Smoothing Splines

A cubic spline smoother is a solution to the following optimization problem: among all functions  $f(x_i)$  with second continuous derivatives, find one that minimizes the penalized least square, which called cubic spline smoother.

$$\sum_{i=1}^n (y_i - f(x_i))^2 + \lambda \int_a^b [f''(x)]^2 dx \quad (2.2.1)$$

where  $\lambda$  is a fixed constant, and  $a \leq x_1 \leq \dots \leq x_n \leq b$ . In the subsequent chapters, I assume  $(a, b)$  includes all possible range.

The name cubic spline is from the piecewise polynomial fit, with the order  $k=3$ , where most study shown that  $k=3$  is sufficient.

Note that the first term represents the least square method. With only this part, the result would be an interpolated curve that would not be smooth at all. It measures closeness to the data while the second term penalizes curvature in the function.  $\int [f''(x)]^2 dx$  measures the ‘wiggleness’ of the function  $f(x)$ . Linear functions have  $\int [f''(x)]^2 = 0$ , while non-linear functions produce non-zero values.

$\lambda$  is a non-negative smoothing parameter that must be chosen by the data analyst. It governs the tradeoff between the goodness of fit to the data and the wiggleness of the function. It plays nearly the same role as span (the percentage of data points used as nearest neighbors in percent of total  $n$ ) in other smoothing methods. When  $\lambda \rightarrow \infty$ , the penalty term becomes more important, forcing  $f''(x) = 0$ , thus the result is the least square line. When  $\lambda \rightarrow 0$ , the penalty becomes unimportant, thus a solution is the second derivative function. The larger values of  $\lambda$  produce smoother curves while the

smaller values produce more wiggly curves. How to choose the appropriate  $\lambda$  will be discussed in the following section.

In order to see it clearly, I will describe cubic smoothing splines in a simple setting.

Suppose that we have a scatterplot of points  $(x_i, y_i)$  shown in Figure 2.2.1.

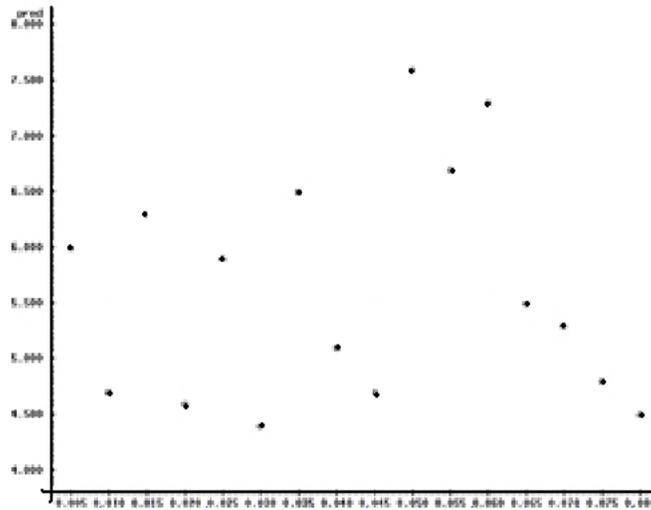


Figure 2.1

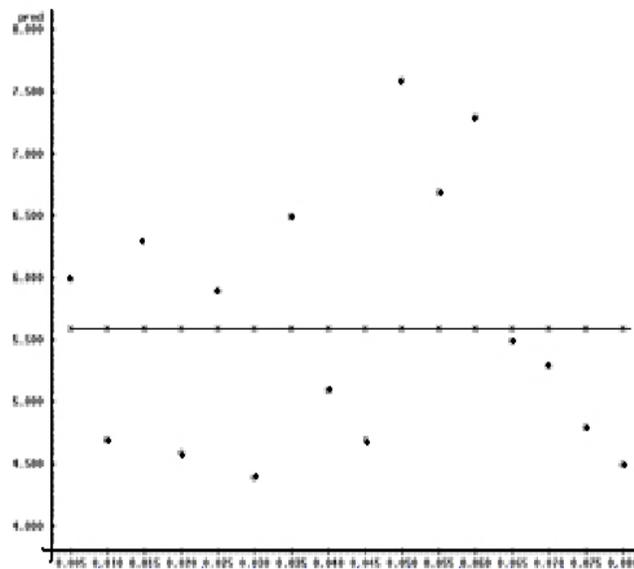


Figure 2.2

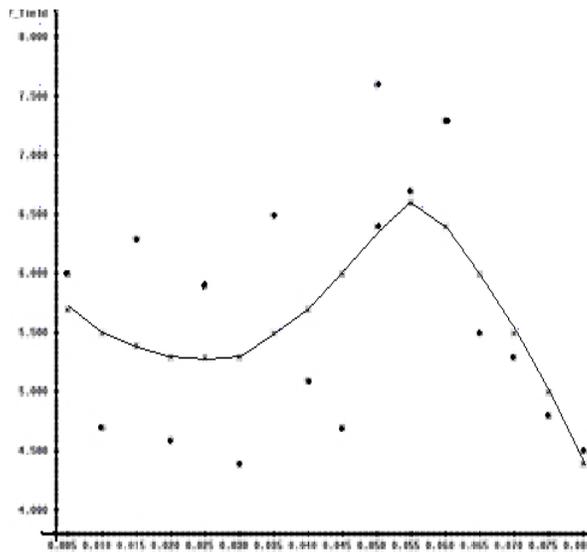


Figure 2.3

Figure 2.2.1 shows a scatterplot of an outcome that measures  $y$  by plotting it against an independent variables  $x$ . In Figure 2.2.2, the straight line was fitted by least square method. In Figure 2.2.3, a cubic spline has been added to describe the trend of  $y$  on  $x$ . Visually, we can see that the cubic smoothing spline describe the trend of  $y$  as a function of  $x$  better than the least square method.

## 2.3 Automatic Selection of Smoothing Parameters $\lambda$

### 2.3.1 From MSE to PSE

In order to minimize the cubic spline smoother, we have to choose the smoothing parameter. To do this, we don't need to minimize the Mean Square Error at each  $x_i$ , but instead we focus on a global measure such as Average Mean Square Error (AMSE)

$$AMSE(\lambda) = \frac{1}{n} \sum_{i=1}^n E \left\{ \hat{f}_\lambda(x_i) - f(x_i) \right\}^2 \quad (2.3.1.1)$$

where  $y_i = f(x_i) + \varepsilon_i$ , and  $\hat{f}_\lambda(x_i)$  is an estimator of  $f(x)$ .

Now we study the Average Predictive Square Error (PSE), which differs from AMSE by only a constant function  $\delta^2$ .

$$PSE(\lambda) = \frac{1}{n} \sum_{i=1}^n E \left\{ Y_i^* - \hat{f}_\lambda(x_i) \right\}^2 \quad (2.3.1.2)$$

where  $Y_i^*$  is a new observation at  $x_i$ , that is  $Y_i^* = f(x_i) + \varepsilon_i^*$ ,  $\varepsilon_i^*$  is independent of  $\varepsilon_i$ 's,

and

$$E(\varepsilon_i^*) = 0.$$

**Lemma 1:**  $PSE = MSE + \sigma^2$ .

$$\text{Since } E \left\{ Y_i^* - \hat{f}_\lambda(x_i) \right\}$$

$$\begin{aligned} &= E \left\{ Y_i^* - \hat{f}_\lambda(x_i) + f(x_i) - f(x_i) \right\}^2 \\ &= E \left\{ Y_i^* - f(x_i) \right\}^2 + E \left\{ \hat{f}_\lambda(x_i) - f(x_i) \right\}^2 + 2E \left\{ Y_i^* - f(x_i) \right\} \left\{ f(x_i) - \hat{f}_\lambda(x_i) \right\} \\ &= E \left\{ Y_i^* - f(x_i) \right\}^2 + E \left\{ \hat{f}_\lambda(x_i) - f(x_i) \right\}^2 + 2E(\varepsilon_i^*)E(\varepsilon_i) \\ &= \delta_i^2 + E \left\{ \hat{f}_\lambda(x_i) - f(x_i) \right\}^2 \end{aligned} \quad (2.3.1.3)$$

$$\begin{aligned}
\text{Then } PSE(\lambda) &= \frac{1}{n} \sum_{i=1}^n E \left\{ Y_i^* - \hat{f}_\lambda(x_i) \right\}^2 \\
&= \frac{1}{n} \sum_{i=1}^n \left( \delta_i^2 + E \left\{ \hat{f}_\lambda(x_i) - f(x_i) \right\}^2 \right) \\
&= \delta^2 + \frac{1}{n} \sum_{i=1}^n E \left\{ \hat{f}_\lambda(x_i) - f(x_i) \right\}^2 \\
&= MSE + \sigma^2 \qquad \text{£ 2.3.1.4}
\end{aligned}$$

Based on the proof, it turned out that one can estimate PSE rather than MSE, which is more appropriate and achievable.

### 2.3.2 Cross Validation

Cross-Validation is the statistical method of partitioning a sample of data into two subsets, a training (calibration) subset for modeling fitting and a test (validation) subset for model evaluation. This approach, however, is not efficient unless the sample is large. The idea behind cross-validation is to recycle data by switching the roles of training and test samples.

During the practice, in order to simplify and not to waste the data set, we are more interested in the Leave-one-out Cross-validation. As the name suggests, the leave-one-out cross-validation (LOOCV) works by leaving the point  $(x_i, y_i)$  out one at a time as the testing set and estimating the smooth at  $x_i$  base on the remaining  $n - 1$  points.

One then constructs the Cross-validation sum of squares. (Sometimes called jackknifed fit at  $x_i$  )

$$CV(\lambda) = \frac{1}{n} \sum_{i=1}^n \left\{ y_i - \hat{f}_\lambda^{-i}(x_i) \right\}^2 \quad (2.3.2.1)$$

where  $\hat{f}_\lambda^{-i}(x_i)$  indicates the fit at  $x_i$  computed by leaving out the  $i^{\text{th}}$  data point.

Using the same idea that was used in the last section,

$$\begin{aligned} & E \left\{ y_i - \hat{f}_\lambda^{-i}(x_i) \right\}^2 \\ &= E \left\{ y_i - f(x_i) + f(x_i) - \hat{f}_\lambda^{-i}(x_i) \right\}^2 \\ &= E \left\{ y_i - f(x_i) \right\}^2 + E \left\{ f(x_i) - \hat{f}_\lambda^{-i}(x_i) \right\}^2 + E \left\{ y_i - f(x_i) \right\} E \left\{ f(x_i) - \hat{f}_\lambda^{-i}(x_i) \right\} \\ &= \delta_i^2 + E \left\{ f(x_i) - \hat{f}_\lambda^{-i}(x_i) \right\}^2 \end{aligned} \quad (2.3.2.2)$$

Here  $E \left\{ y_i - f(x_i) \right\} E \left\{ f(x_i) - \hat{f}_\lambda^{-i}(x_i) \right\} = 0$ , because  $\hat{f}_\lambda^{-i}(x_i)$  doesn't involve  $y_i$ .

Moreover if, we assume  $\hat{f}_\lambda^{-i}(x_i) \approx \hat{f}_\lambda(x_i)$ ,

$$\begin{aligned}
& E\{CV(\lambda)\} \\
&= E\left\{\frac{1}{n} \sum_{i=1}^n \left\{y_i - \hat{f}_{\lambda}^{-i}(x_i)\right\}^2\right\} \\
&= \delta^2 + E\left\{f(x_i) - \hat{f}_{\lambda}^{-i}(x_i)\right\}^2 \\
&= \delta^2 + E\left\{\hat{f}_{\lambda}(x_i) - f(x_i)\right\}^2 \\
&= PSE
\end{aligned} \tag{2.3.2.3}$$

Since minimizing  $CV(\lambda)$  is equivalent to minimizing  $PSE(\lambda)$ , we can use  $CV(\lambda)$  for smoothing parameter selection. A suitable range of  $\lambda$  will be used to compute  $CV(\lambda)$ , the  $\lambda$  value which minimizes  $CV(\lambda)$  will be selected as the smoothing parameter.

### 2.3.3 Generalized Cross-validation

To compute the  $CV(\lambda)$  score, it seems that one needs to fit model  $n$  times, once for each leave-one-out data  $y^{-i}$ . This is computationally intensive. However with some algebra, a shortcut exists for many situations.

**Lemma 2:** 
$$CV(\lambda) = \frac{1}{n} \sum_{i=1}^n \left\{ \frac{y_i - \hat{f}_{\lambda}(x_i)}{1 - s_{ii}} \right\}^2$$

Since  $\hat{f} = S_{\lambda}y$ , where  $S = \{s_{ij}\}$  is a  $n \times n$  matrix called the smoother matrix when we use the leaving-out-one method.

Let

$$\tilde{y}_{ij} = \begin{cases} y_j & j \neq i \\ \hat{f}_\lambda^{-i}(x_i) & j = i \end{cases} \quad (2.3.3.1)$$

Then

$$\hat{f}_\lambda(x_i) = \sum_{j=1}^n s_{ij} y_j \quad (2.3.3.2)$$

$$\hat{f}_\lambda^{-i}(x_i) = \sum_{j=1}^n s_{ij} \tilde{y}_{ij} = \sum_{j \neq i} s_{ij} y_j + s_{ii} \hat{f}_\lambda^{-i}(x_i) \quad (2.3.3.3)$$

Combine Equations (2.3.3.2) and (2.3.3.3) we have

$$\hat{f}_\lambda(x_i) - \hat{f}_\lambda^{-i}(x_i) = s_{ii} y_i - s_{ii} \hat{f}_\lambda^{-i}(x_i)$$

$$\Rightarrow (s_{ii} - 1) \hat{f}_\lambda^{-i}(x_i) = s_{ii} y_i - \sum_{j \neq i} s_{ij} y_j$$

$$\Rightarrow \hat{f}_\lambda^{-i}(x_i) = \frac{\sum_{j=1, j \neq i} s_{ij} y_j}{1 - s_{ii}} \quad (2.3.3.4)$$

Then

$$y_i - \hat{f}_\lambda^{-i}(x_i) = \frac{(1 - s_{ii})y_i - \sum_{j=1, j \neq i} s_{ij} y_j}{1 - s_{ii}}$$

$$= \frac{y_i - s_{ii} y_i - \sum_{j=1, j \neq i} s_{ij} y_j}{1 - s_{ii}}$$

$$= \frac{y_i - \hat{f}_\lambda(x_i)}{1 - s_{ii}} \quad (2.3.3.5)$$

Thus

$$CV(\lambda) = \frac{1}{n} \sum_{i=1}^n \left\{ \frac{y_i - \hat{f}_\lambda(x_i)}{1 - s_{ii}} \right\}^2. \quad (2.3.3.6)$$

Thus the fit  $\hat{f}_\lambda^{-i}(x_i)$  can be computed from  $\hat{f}_\lambda(x_i)$  and  $s_{ii}$ . There is no need to actually remove the  $i^{\text{th}}$  point and re-compute the smooth. We only need to fit model once with the full data and compute the diagonal elements of the smother matrix.

If we replace  $s_{ii}$  by the averages of all the diagonal elements, it results in the following Generalized Cross Validation (GCV).

$$GCV(\lambda) = \frac{1}{n} \sum_{i=1}^n \left\{ \frac{y_i - \hat{f}_\lambda(x_i)}{1 - \frac{trS}{n}} \right\}^2 \quad (2.3.3.7)$$

GCV is a weighted version of CV with weights  $\frac{(1 - s_{ii})^2}{(1 - \frac{trS}{n})^2}$ .

If  $\frac{trS}{n}$  is small, using the approximation  $(1 - x)^{-2} \approx 1 + 2x$ , then

$$\begin{aligned} GCV(\lambda) &\approx \frac{1}{n} \sum_{i=1}^n \left\{ y_i - \hat{f}_\lambda(x_i) \right\}^2 * \left\{ 1 + \frac{1}{n} \cdot 2 \cdot trS \right\} \\ &= \frac{1}{n} \sum_{i=1}^n \left\{ y_i - \hat{f}_\lambda(x_i) \right\}^2 + \frac{2}{n} trS \left[ \frac{1}{n} \sum_{i=1}^n \left\{ y_i - \hat{f}_\lambda(x_i) \right\}^2 \right] \end{aligned} \quad (2.3.3.8)$$

If we regard  $\frac{1}{n} \sum_{i=1}^n \left\{ y_i - \hat{f}_\lambda(x_i) \right\}^2$  in the second part as an estimate of  $\hat{\sigma}^2$ , the GCV is

approximately the same as  $C_p$  statistics.

$$GCV(\lambda) = ASR + 2trS * \frac{\hat{\sigma}^2}{n} \approx C_p \text{ Statistics} \quad (2.3.3.9)$$

Thus GCV can be used to obtain the minimal smoothing parameter. Show below is an example of how the smoothing parameter is chosen by GCV function using SAS.

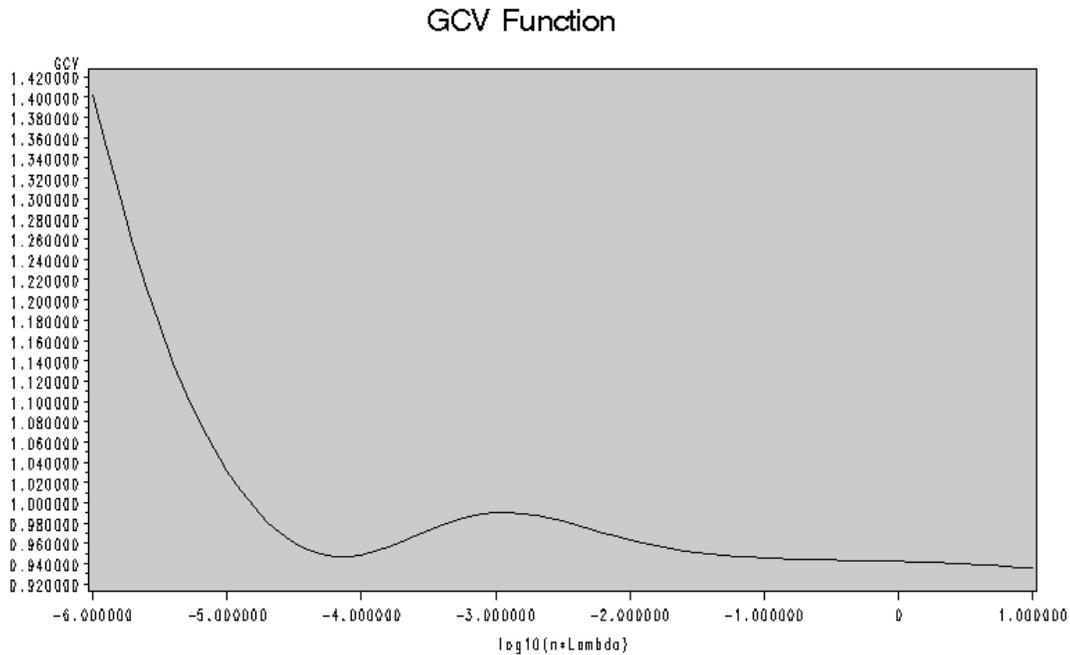


Figure 2.4

Figure 2.4 is the plot of the GCV function vs.  $\log_{10} n\lambda$ . The GCV function has two minima. The plot shows a minimum at 1.1. The figure also suggests a local minimum around -4.2. Note that the SAS procedure for GCV function may not always find the global minimum, though it did in this case. We can calculate the corresponding  $\lambda$  to be about 0.11. In this case the degree of freedom is about 8.8.

### 2.3.4 Degrees of Freedom of a Smoother

In fact it is inconvenient to express the desired smoothness of functions in terms of  $\lambda$ . In the SAS GAM procedure, we also select the value of a smoothing parameter simply by specifying the *df* for the smoother.

Degrees of freedom of a smoother ( $df(Smother)$ ), sometimes called effective number of parameters, is an indication of the amount of smoothing. Given a linear smoother  $S_\lambda$ , the degrees of freedom  $df$  is defined as  $df(Smother) = tr(S_\lambda)$ . The smoothing parameter is the major determinant of the degrees of freedom. More smoothing means fewer  $df(Smother)$  or higher span. The  $df(Smother)$  is not necessarily an integer.

DF	Lambda
2.5	1
4	0.5
5	0.3
6	0.22
10	0.10

Lots of smoothing  
↑

Table 2.1 Approximate correspondence between Degree of freedom and span.

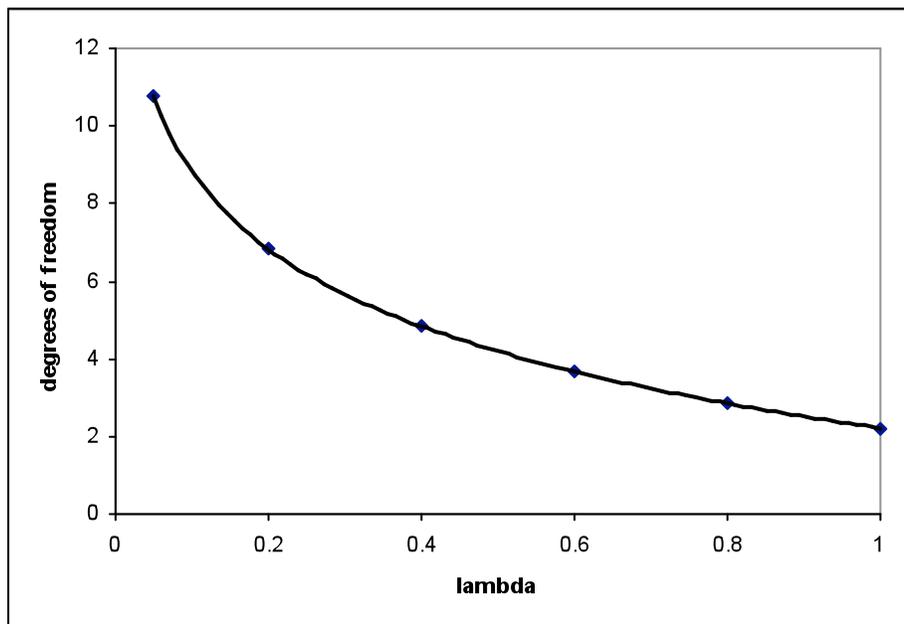


Figure 3.6 Approximate correspondence between Degree of freedom and span.

## Chapter 3 Generalized Additive Model (GAM)

### 3.1 Generalized Additive Model (GAM)

The generalized additive model (GAM) is a statistical model developed by Hastie and Tibshirani in the year 1990. It is applicable in many areas of prediction.

An additive model has the form

$$Y = f_0 + \sum_{j=1}^p f_j(X_j) + \varepsilon \quad (3.1.1)$$

where the errors  $\varepsilon$  are independent of the  $X_j$ 's,  $E(\varepsilon) = 0$  and  $Var(\varepsilon) = \sigma^2$ . The  $f_j$ 's are arbitrary univariate functions, one for each variable.

### 3.2 Fitting the Generalized Additive Model

There are a number of approaches for the formulation and estimation of additive models. The back-fitting algorithm is a general algorithm that can fit an additive model using any regression-type fitting mechanism.

Define the  $j^{\text{th}}$  set of partial residuals as

$$R_j = Y - f_0 - \sum_{k \neq j} f_k(X_k) \quad (3.2.1)$$

then  $E(R_j | X_j) = f_j(X_j)$ . This observation provides a way for estimating each

smoothing function  $\hat{f}_j$ .

Given an observation  $(x_i, y_i)$ , a criterion like the penalized sum of squares can be specified for this problem, which is

$$\sum_{j=1}^p (y_j - f_0 - f_j(x_j))^2 + \sum_{j=1}^p \lambda_j \int [f_j''(t_j)]^2 dt_j \quad (3.2.2)$$

Algorithm: The back-fitting algorithm for additive models

1. Initialize:

$$\hat{f}_0 = \frac{1}{n} \sum_{i=1}^n y_i, \quad \hat{f}_j \equiv 0, \quad m=1$$

2. Iterate:

$$m \leftarrow m + 1, \quad j=1, 2, \dots, p$$

$$f_j^m(x_j) \leftarrow S_j \left[ \left\{ y_i - \hat{f}_0 - \sum_{k=1}^{j-1} \hat{f}_k^m(x_k) - \sum_{k=j+1}^p \hat{f}_k^{m-1}(x_k) \right\}_1^n \right] \quad (3.2.3)$$

3. Until:

$$RSS = S(Y - \hat{f}_0 - \sum_{k=1}^p \hat{f}_k^m(x_k))^2 \text{ fails to decrease, which means the function}$$

$\hat{f}_j$  changes less than a prespecified threshold.

$f_j^m(\cdot)$  denotes the estimate of  $f_j(\cdot)$  at the  $m^{\text{th}}$  iteration. Each of the functions  $f_j$  is a cubic spline in the component  $x_j$ , with knots at each of the unique values of  $x_{ij}$ ,

$j = 1, 2, \dots, p$ .

The GAM procedure uses the following condition as the convergence criterion for the backfitting algorithm:

$$\frac{\sum_{j=1}^p \sum_{i=1}^n (f_j^{m-1}(x_{ij}) - f_j^m(x_{ij}))^2}{\sum_{j=1}^p \sum_{i=1}^n f_j^{m-1}(x_{ij})^2} \leq \varepsilon \quad (3.2.4)$$

where  $\varepsilon = 10^{-8}$  by default.

Now let's take a look at the Back-fitting Algorithm in details.

When  $m = 1 + 1 = 2, j = 1$

$$\begin{aligned} f_1^2(x_1) &= S_1 \left[ \left\{ y_i - \hat{f}_0 - \sum_{k=1}^{l-1} \hat{f}_1^2(x_1) - \sum_{k=2}^p \hat{f}_2^1(x_k) \right\}_1^n \right] \\ &= S_1 \left[ \left\{ y_i - \frac{1}{n} \sum_{i=1}^n y_i \right\}_1^n \right] \end{aligned} \quad (3.2.5)$$

where  $\sum_{k=1}^{l-1} \hat{f}_1^2(x_1) = \sum_{k=1}^0 \hat{f}_1^2(x_1)$  does not exist, since the upper limit is smaller than the lower limit.

By applying a cubic smoothing spline  $S_1$  to the targets  $\left\{ y_i - \hat{f}_0 \right\}$ , as a function of  $x_{i1}$ , we

will obtain a new estimate  $\hat{f}_1^2(x_1)$ .

When  $j = 2$

$$\begin{aligned}
 f_2^2(x_2) &= S_2 \left[ \left\{ y_i - \hat{f}_0 - \sum_{k=1}^{2-1} \hat{f}_1^2(x_1) - \sum_{k=3}^p \hat{f}_3^1(x_k) \right\}_1^n \right] \\
 &= S_2 \left[ \left\{ y_i - \frac{1}{n} \sum_{i=1}^n y_i - \hat{f}_1^2(x_{i1}) \right\}_1^n \right] \tag{3.2.6}
 \end{aligned}$$

By repeating this, we can estimate  $\hat{f}_j^2(x_j)$ , all of the smoothing functions at the 2<sup>nd</sup> iteration.

Then goes to next iteration,  $m = 2 + 1 = 3. j = 1$

$$f_1^3(x_1) = S_1 \left[ \left\{ y_i - \hat{f}_0 - \sum_{k=1}^{1-1} \hat{f}_1^3(x_1) - \sum_{k=2}^p \hat{f}_k^2(x_k) \right\}_1^n \right] \tag{3.2.7}$$

where  $\sum_{k=2}^p \hat{f}_k^2(x_k)$  has already been found in the last cycle,  $\sum_{k=1}^{1-1} \hat{f}_1^3(x_1) = \sum_{k=1}^0 \hat{f}_1^3(x_1)$  does not exist.

By doing the same thing as  $m = 2$  in the 2<sup>nd</sup> iteration, this time we will find the  $\hat{f}_1^3(x_1)$ .

Keep iterating and we will find all the smoothing functions.

### 3.3 The Generalized Additive Logistic Model

For binary data, the most popular approach is logistic regression,

let  $y = \begin{cases} 1 & \text{with probability } p(x) \\ 0 & \text{with probability } 1 - p(x) \end{cases}$  where  $X = (X_1, \dots, X_p)$  is a vector of covariates.

$$\log \text{it}(p(X)) = \log \frac{p(X)}{1 - p(X)} = \beta_0 + \sum_{j=1}^p \beta_j(x_{ij})$$

$$\text{and } p(X) = \frac{\text{Exp}(\beta_0 + \sum_{j=1}^p \beta_j(x_{ij}))}{1 + \text{Exp}(\beta_0 + \sum_{j=1}^p \beta_j(x_{ij}))} \quad (3.3.1)$$

In logistic GAM, the basic idea is to replace the linear predictor with an additive one:

$$\log \text{it}(p(X)) = \log \frac{p(X)}{1 - p(X)} = f_0 + \sum_{j=1}^p f_j(x_{ij})$$

$$\text{and } p(X) = \frac{\text{Exp}(f_0 + \sum_{j=1}^p f_j(x_{ij}))}{1 + \text{Exp}(f_0 + \sum_{j=1}^p f_j(x_{ij}))} \quad (3.3.2)$$

Generally, let  $E(Y | X) = \mu$ , ( $\mu_i = 1 * p_i + 0 * (1 - p_i) = p_i$ ).

$$\eta(x) = g(\mu) = \log \frac{p(x)}{1 - p(x)} \quad (3.3.3)$$

where  $\eta$  is a function of  $p$  variables.

Assume  $Y = \eta(x) + \varepsilon$ , given some initial estimate of  $\eta(x)$ , construct the adjusted dependent variable

$$Z_i = \eta_i + (y_i - \mu_i) \left( \frac{\partial \eta_i}{\partial \mu_i} \right) \quad (3.3.4)$$

Instead of fitting an additive model to  $Y$ , we fit an additive model to the  $Z_i$ 's, treating it

as the response variable  $Y$  in  $E(Y | X) = f_0 + \sum_{j=1}^p f_j(x_{ij})$ .

The algorithm is to fit the smoothing functions that is similar to the algorithm described earlier.

Algorithm: General Local Scoring Algorithm

1. Initialize:

$$f_0 = g\left(\sum_{i=1}^n \frac{y_i}{n}\right), f_1^0 = \dots = f_p^0 = 0$$

2. Iterate:

$$m = m + 1$$

from the adjusted dependent variable

$$Z_i = \eta_i + (y_i - \mu_i) \left( \frac{\partial \eta_i}{\partial \mu_i} \right) \quad (3.3.5)$$

$$\text{where } \eta^{m-1} = f_0 + \sum_{j=1}^p f_j^{m-1}(x_{ij})$$

$$\eta^{m-1} = g(\mu^{m-1}) \quad (3.3.6)$$

$$\text{so } \mu^{m-1} = g^{-1}(\eta_i) \quad (3.3.7)$$

$$\text{construct weights } w_i = \left( \frac{\partial \eta_i^{m-1}}{\partial \mu_i^{m-1}} \right)^2 v_i^{-1}, v_i = \text{Var}(Y_i)$$

Fit a weighted additive model to  $Z_i$  to obtain estimated function  $f_j^m$ , additive predictor  $\eta^m$ , and fitted value  $\mu_i^m = p_i$

3. Until:

Repeat step 2, until the change deviance is sufficiently small.

$$D = 2 \left\{ \log(L(Y, p) - \log(L(Y, \hat{P}^{LM})) \right\}$$

$$= 2 \left\{ \log(L(Y, \mu) - \log(L(Y, \hat{\mu}^{LM})) \right\} \quad (3.3.8)$$

$$\text{or compute } \Delta(\eta^m, \eta^{m-1}) = \frac{\sum_{j=1}^p \|f_j^m - f_j^{m-1}\|}{\sum_{j=1}^p \|f_j^{m-1}\|} \quad (3.3.9)$$

The GAM procedure uses the following condition as the convergence criterion for the back-fitting algorithm:

$$\frac{\sum_{j=1}^p \sum_{i=1}^n w(x_i) (f_j^{m-1}(x_{ij}) - f_j^m(x_{ij}))^2}{\sum_{j=1}^p \sum_{i=1}^n w(x_i) f_j^{m-1}(x_{ij})^2} \leq \varepsilon \quad (3.3.10)$$

where  $\varepsilon = 10^{-8}$  by default.

Given some initial estimations of  $\eta(x)$ , a first order Taylor series expansion, together with the fisher scoring method, will give the improved estimate

$$\eta^{est}(x) = \eta^{given}(x) + \delta \quad (3.3.11)$$

$$\begin{aligned}
\delta &= \frac{\text{score function}}{\text{Expected information matrix}} \\
&= \frac{dL/d\eta}{E(-d^2L/d\eta^2 | x)} \\
&= E(\eta(x) - \frac{dL/d\eta}{E(-d^2L/d\eta^2 | x)} | x) \tag{3.3.12}
\end{aligned}$$

$$\frac{dL}{d\eta} = \frac{dL}{d\mu} \cdot \frac{d\mu}{d\eta} \tag{3.3.13}$$

$$L = \ln \prod_{i=1}^p \mu_i^{y_i} (1 - \mu_i)^{1-y_i} \tag{3.3.14}$$

$$\begin{aligned}
\frac{\partial L}{\partial \mu_i} &= \mu_i \cdot \frac{1}{\mu_i} + (1 - y_i) \frac{-1}{1 - \mu_i} \\
&= \frac{(1 - \mu_i)y_i - \mu_i(1 - y_i)}{\mu_i(1 - \mu_i)} \\
&= \frac{y_i - \mu_i}{(1 - \mu_i)\mu_i} \tag{3.3.15}
\end{aligned}$$

Since  $\text{Var}Y_i = EY_i^2 - (EY_i)^2 = 1^2 \cdot \mu_i + 0^2 \cdot (1 - \mu_i) - \mu_i = \mu_i(1 - \mu_i)$  and  $v_i^{-1} = \frac{1}{\mu_i(1 - \mu_i)}$

$$\text{so } \frac{dL}{d\eta} = (y - \mu) \cdot v^{-1} \cdot \frac{d\mu}{d\eta} \tag{3.3.16}$$

$$\frac{d^2L}{d\eta^2} = (y - \mu) \cdot \frac{d}{d\eta} \cdot (v^{-1} \frac{d\mu}{d\eta}) - (\frac{d\mu}{d\eta})^2 v^{-1} \tag{3.3.17}$$

$$\text{so } E(d^2L/d\eta^2 | x) = -(\frac{d\mu}{d\eta})^2 v^{-1} \tag{3.3.18}$$

$$\eta^{est}(x) = E \left[ \eta(x) + (Y - \mu) \cdot \frac{d\eta}{d\mu} \mid x \right] \quad (3.3.19)$$

Then replace the conditional estimations by smoothers

$$\eta^{est}(x) = smoother \left[ \eta(x) + (Y - \mu) \cdot \frac{d\eta}{d\mu} \mid x \right] \quad (3.3.20)$$

## Chapter 4 Examples

### 4.1 Example 1: Comparing the GAM and the GLM

In SAS, the PROC GAM procedure in SAS implements the Generalized Additive Model and the PROC GLM procedure fits the Generalized Linear Model.

In an analysis of a simulation from a chemistry experiment, Generalized Additive Model fitted by the PROC GAM procedure is compared to the Generalized Linear Model fitted by using the PROC GLM procedure.

The data set consists of 3 variables. They are measurements of the temperature of the experiment, the amount of catalyst added, and the yield of the chemical reaction. Here the yield is the response variable.

Temperature	Catalyst	Yield
80	0.005	6.039
80	0.010	4.719
80	0.015	6.301
...	...	...
90	0.005	4.540
90	0.010	3.553
...	...	...
140	0.080	6.006

Table 4.1: Sample data set for example 1.

The following plots were generated by SAS.

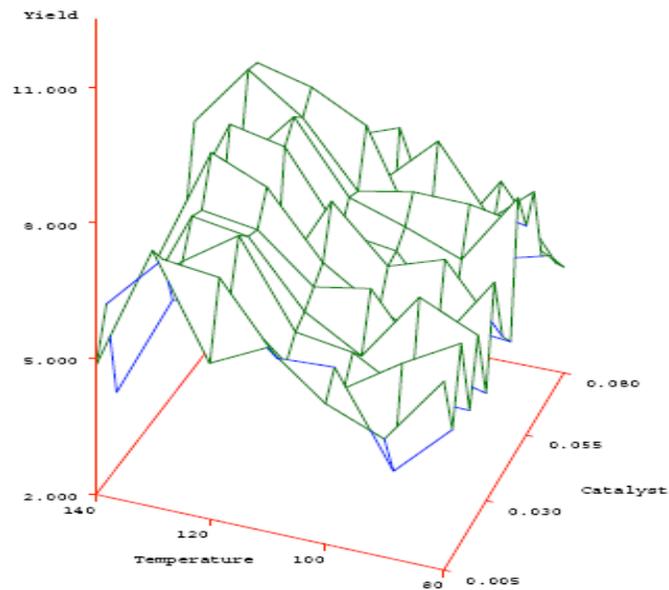


Figure 4.1 Plot of the raw data.

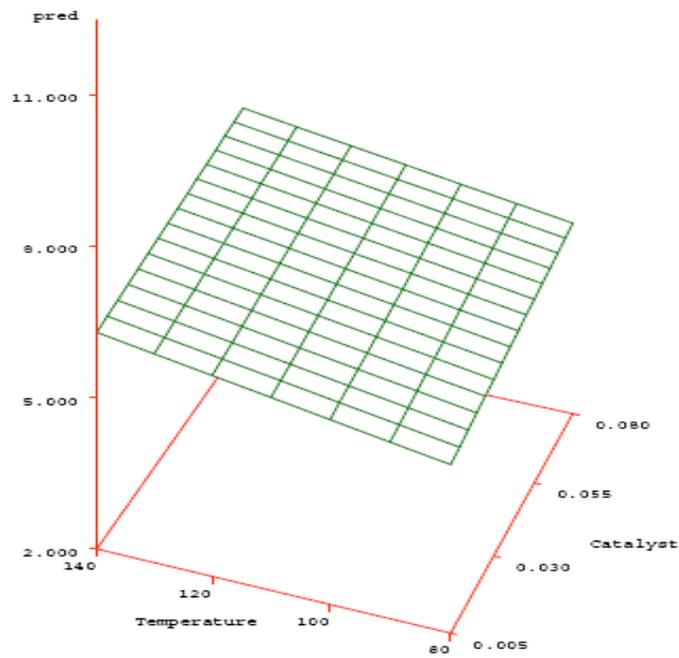


Figure 4.2 The predicted plane fitted by the GLM procedure.

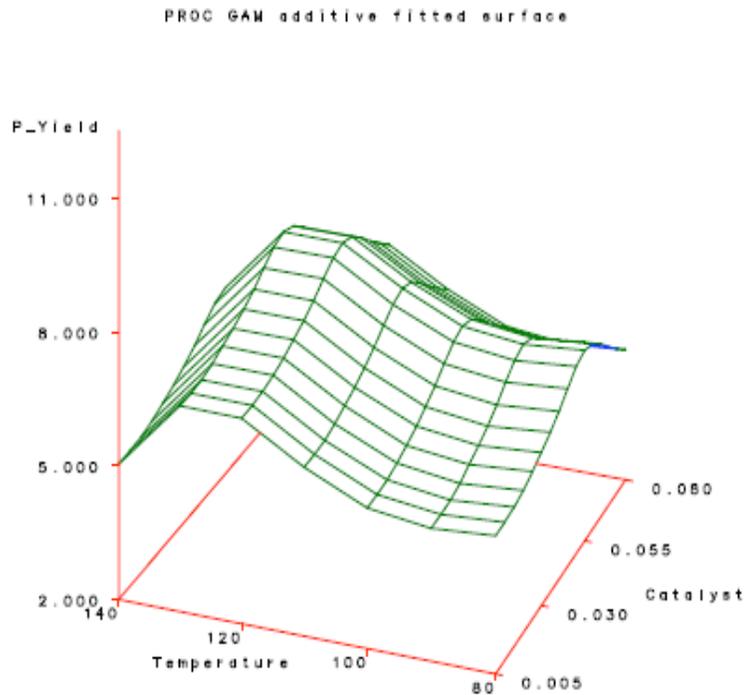


Figure 4.3 The predicted surface fitted by the GAM procedure.

Figure 4.1 is the surface plot of Yield by Temperature and Amount of Catalyst of the raw data. Figure 4.2 is the plane fitted by using the PROC GLM procedure. Figure 4.3 is the surface fitted by the PROC GAM. Clearly the predicted surface using the GAM procedure produces much better prediction results.

#### 4.2 Example 2: Fitting Logistic GAM Model using the GAM Procedure

The data used in this example are based on a study of chronic bronchitis in Cardiff conducted by Jones(1975). The data in the study consist of observations on three variables for each of 212 men in a Cardiff districts sample. The variables are CIG, the number of cigarettes smoked per day, POLL, the smoke level in the locality of the respondent's home (obtained by interpolation from 13 air pollution monitoring stations on the city), and R, an indicator variable taking the value 1 if the respondent suffered from chronic bronchitis, and 0 if he did not.

#### Data Description

R Chronic Bronchitis (1=presence, 0=absence)	CIG Cigarette Consumption	POLL Smoke Level of Locality of Respondents Home
0	5.15	67.1
1	0.0	66.9
0	2.5	66.7
0	1.75	65.8

0	6.75	64.4
0	0.0	64.4
...	...	...
0	0.9	55.4

Table 4.2: Sample data set for example 2.

The first step of data exploration is to visualize the distribution of numeric predictors. Empirical distribution of variables CIG and POLL were estimated with kernel density estimation using the KDE procedure in SAS and were shown in Figure 4.4.

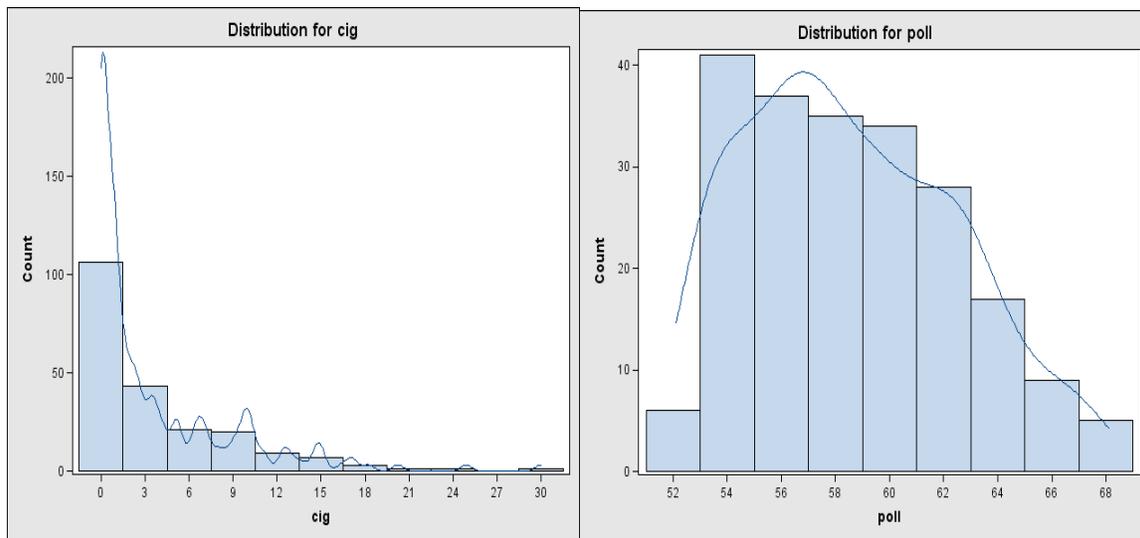


Figure 4.4 Distributions of CIG and POLL

Clearly, both numeric predictors are skewed to the left, as shown in Figure 4.4.

The correlations between numeric predictors are also investigated using the CORR procedure in SAS and the result is presented in Figure 4.5 and Figure 4.6.

Pearson Correlation Coefficients, N = 212 Prob >  r  under H0: Rho=0		
	cig	poll
cig	1.00000	0.08170 0.2362
poll	0.08170 0.2362	1.00000

Figure 4.5 Correlation Matrix of CIG and POLL

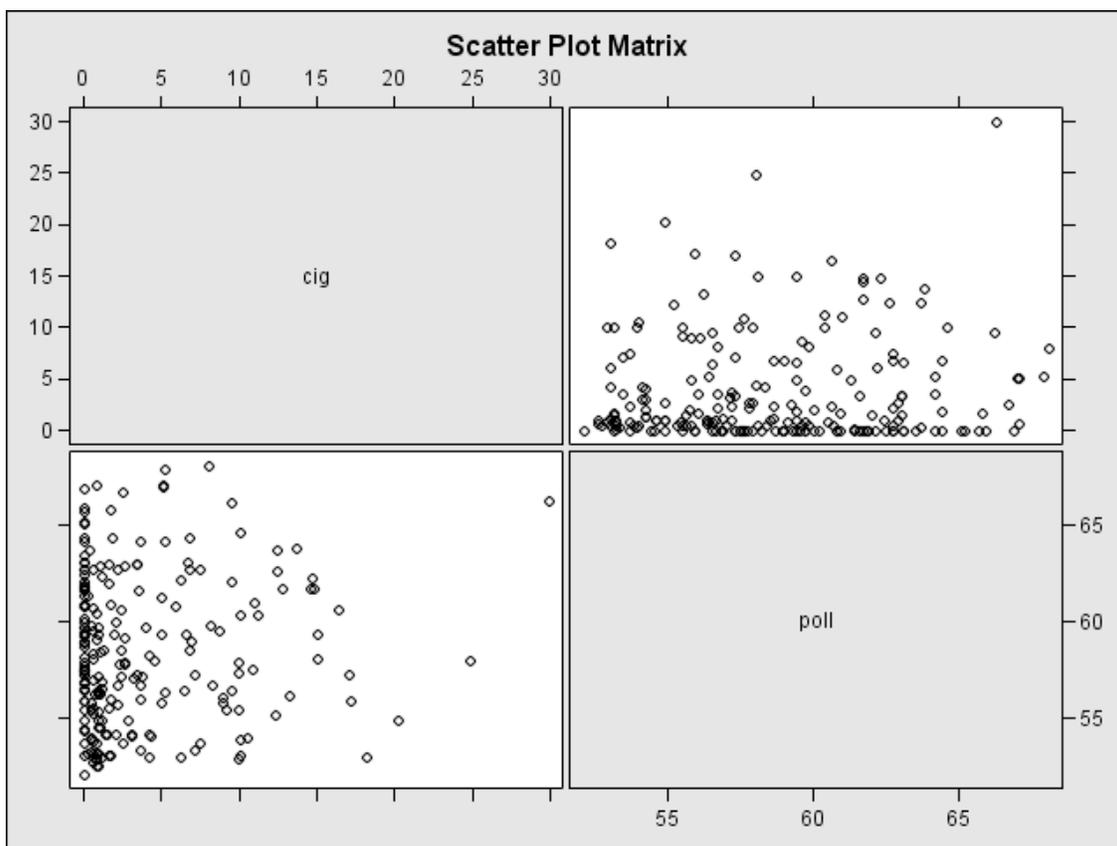


Figure 4.6 Scatter Plot Matrix of CIG and POLL

The symmetric plot suggests no major correlation between variables. Therefore, the impact of multicollinearity on parameter estimates is not a major concern.

## Modeling

PROC GAM procedure was then used to fit a model, with flexible spline terms for each of the predictors. In this model, we choose  $df = 3$ . Of these three degrees of freedom, one is taken up by the linear portion of the fit and the other two are left for the nonlinear spline portion. Although this might seem to be a modest amount of flexibility, it is better to use this number with a small data set. This avoids over-fitting and lower the computing cost.

Parts of the SAS outputs from the PROC GAM procedure are listed in Figures 4.7, 4.8, 4.9 and Figure 4.10.

**The SAS System**  
*The GAM Procedure*  
*Dependent Variable: bronchitis*  
*Smoothing Model Component(s): spline(cig) spline(poll)*

Summary of Input Data Set	
Number of Observations	212
Number of Missing Observations	0
Distribution	Binomial
Link Function	Logit

Iteration Summary and Fit Statistics	
Number of local score iterations	6
Local score convergence criterion	1.440765E-10
Final Number of Backfitting Iterations	1
Final Backfitting Criterion	2.239605E-10
The Deviance of the Final Estimate	157.01533552

Figure 4.7 Summary for the input data and algorithms

Regression Model Analysis Parameter Estimates				
Parameter	Parameter Estimate	Standard Error	t Value	Pr >  t
Intercept	-9.72547	3.09992	-3.14	0.0020
Linear(cig)	0.18624	0.03485	5.34	<.0001
Linear(poll)	0.12850	0.05158	2.49	0.0135

Smoothing Model Analysis Fit Summary for Smoothing Components				
Component	Smoothing Parameter	DF	GCV	Num Unique Obs
Spline(cig)	0.999976	2.000000	6.824160	94
Spline(poll)	0.999994	2.000000	12.931922	110

Figure 4.8 Analytical information about the fitted model

Smoothing Model Analysis Analysis of Deviance				
Source	DF	Sum of Squares	Chi-Square	Pr > ChiSq
Spline(cig)	2.00000	15.964621	15.9646	0.0003
Spline(poll)	2.00000	1.445150	1.4451	0.4855

Figure 4.9 Analysis of Deviance

The first part of the output from the PROC GAM procedure (Figure 4.7) summarizes the input data set and provides a summary for the bac-kfitting and local scoring algorithms. The second part of the output (Figure 4.8) provides the analytical information about the fitted model.

The critical part of the output is the “Analysis of Deviance” Table, shown in Figure 4.7. For each smoothing effect in the model, this table gives a Chi-square test comparing the

deviance between the full model and the model without this variable. In this case, the analysis of deviance results indicated that the effects of CIG is significant, while POLL is insignificant at 5% level.

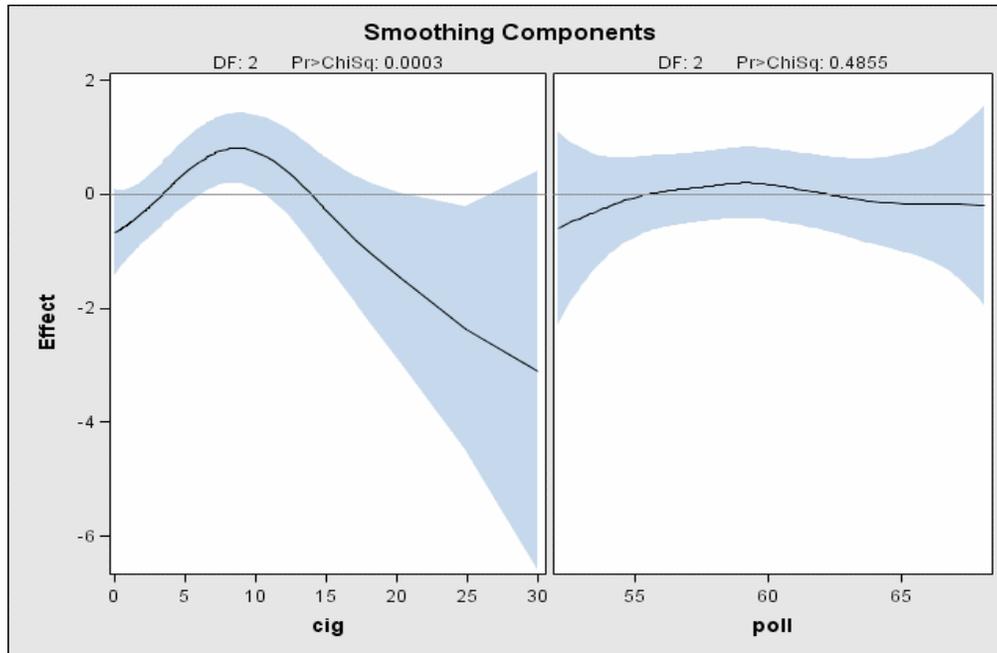


Figure 4.8 Partial Prediction for Each Predictor

The plots show that the partial predictions corresponding to both CIG and POLL have a quadratic pattern. Note that the 95% confidence limits for the Poll contains the zero axis, confirming the insignificance of this term.

The GAM procedure generated the following output (Figure 4.9), including the predicted logit and predicted probability. The predicted logit is not of direct interest. Use inverse link function to convert logit to probability. For example, when the number of cigarettes smoked per day is 5.15 and the smoke level in the locality of the respondent's home equals 67.1, according to the Logistic GAM estimated from the data, the probability of suffering from chronic bronchitis is 0.51881.

Obs	Obs	bronchitis	cig	poll	P_bronchitis	phat
1	1	0	5.15	67.1	0.07527	0.51881
2	2	1	0.00	66.9	-1.98438	0.12085
3	3	0	2.50	66.7	-1.07199	0.25502
4	4	0	1.75	65.8	-1.48408	0.18481
5	5	0	6.75	64.4	0.33496	0.58297
6	6	0	0.00	64.4	-2.27941	0.09284
7	7	1	0.00	65.1	-2.20296	0.09948
8	8	1	9.50	66.2	1.16294	0.76187
9	9	0	0.00	65.9	-2.10503	0.10861
10	10	0	0.75	67.1	-1.70070	0.15437
11	11	0	5.25	67.9	0.20041	0.54994
12	12	1	8.00	68.1	1.10961	0.75206

Figure 4.9 Partial output of Logistic Generalized Additive Model  
P\_bronchitis is the predicted logit. Phat represents the predicted probability.

## **Chapter 5 Discussion**

Generalized additive models provide a flexible method for uncovering nonlinear relationship between dependent and independent variables in exponential family and other likelihood based regression models. In the two data examples given in this paper, the iterative method called the back-fitting algorithm was used to predict the unspecified function. The cubic spline smoother was applied to the independent variables.

## References:

1. Hasties, T.J. and Tibshirani, R.J. (1990), Generalized Additive MODELS, New York: Chapman and Hall.
2. Friedman, J., Hasties, T.J. and Tibshirani, R.J. (2000), Additive logistic regression: a statistical view of boosting, The Annals of Statistics 2000, Vol. 28, No. 2,337-407.
3. Dong, X. (2001), Fitting Generalized Additive Models with the GAM procedure, SUGI Proceedings.
4. Wood, Simon N. Generalized Additive Model: an introduction with R, Chapman and Hall/CRC.
5. Friedman, J. (1991), Multivariate adaptive regression splines, Annals of Statistics 19(1), 1-141.
6. Marx, Brain D. and Eilers, Paul H.C (1998), Direct generalized additive modeling with pernalized likelihood, Computational Statistics and Data Analysis 28(1998) 193-209.
7. Liu, Wensui and Cela, Jimmy (2007), Improving credit scoring by generalized additive model, SAS global forum 2007 (paper 078-2007)
8. [http://en.wikipedia.org/wiki/Generalized\\_additive\\_model](http://en.wikipedia.org/wiki/Generalized_additive_model)

## Appendix SAS Code

### A. SAS Code Used to Choose the Smoothing Parameter by GCV.

```
data bronchitis;
infile 'bronchitis.dat' ;
input bronchitis cig poll @@;
run;
proc tpspline data=bronchitis;
ods output GCVFunction=gcv;
model bronchitis = (cig poll) /lognlambd=(-6 to 1 by 0.1);
run;
symbol1 interpol=join value=none;
title "GCV Function";
proc gplot data=gcv;
plot gcv*lognlambd/frame cframe=ligr
vaxis=axis1 haxis=axis2;
run;
```

## B. SAS Code Used to compare PROC GAM with PROC GLM.

```
*****
*1.This SAS program use PROC G3D to fit the surface plot of the raw*
*data set. The raw data set include three numeric variables: two  *
*horizontal variables plotted on the x and y axes the difine an x-y*
*plane, and a vertical variable plotted on the z axis rising from  *
*the {it x-y}plane.                                             *
*2.Fit models with PROC GLM and PROC GAM.                       *
*3.Plot predict yield value from both GLM and GAM models agaist  *
* temperature and catalystr.                                     *
*****
```

```
ods pdf;
ods graphics on;

libname here '.';

data ExperimentA;
  infile 'ExperimentA.dat;
  format Temperature f4.0 Catalyst f6.3 Yield f8.3;
  input Temperature Catalyst Yield @@;

ods graphics on;

title2 'Raw data';
proc g3d data=ExperimentA;
  plot Temperature*Catalyst=Yield
  / zmin=2 zmax=11;
run;

proc glm;
  model Yield=Temperature Catalyst;
  output out=here.a p=pred;
run

Proc Print data=here.a;
run;
```

```

proc g3d data=here.a;
  format pred f6.3;
  plot Temperature*Catalyst=pred
    / name= 'GLM' zmin=2 zmax=11;
run;

proc gam data=ExperimentA;
  model Yield = loess(Temperature) loess(Catalyst)
    / method=gcv;
  output out=PredGAM;
run;

proc g3d data=PredGLM;
  format P_Yilde f6.3;
  plot Temperature*Catalyst=P_Yield
    / name= 'GLM' zmin=2 zmax=11;
run;

goptions display;
proc greplay nofs tc=sashelp.templt template=v2;
  igout=gseg;
  treplay 1:GAMA 2:GLM;
run;
quit;

ods graphics off;
ods pdf close;

```

### C. SAS Code Used to fit Logistic GAM model using PROC GAM procedure.

```

ods html;
ods graphics on;

/* Input data set*/
data bronchitis;
    input bronchitis cig poll @@;
    datalines;
0 5.15 67.1 1 0.0 66.9 0 2.5 66.7 0 1.75 65.8
0 6.75 64.4 0 0.0 64.4 1 0.0 65.1 1 9.5 66.2
0 0.0 65.9 0 0.75 67.1 0 5.25 67.9 1 8 68.1
1 5.15 67 1 30.0 66.3 0 0.0 65.7 0 0.0 65.2
0 5.25 64.2 0 10.05 64.6 0 0.0 63.5 1 3.4 63.0
0 0.0 62.7 0 .55 62.7 1 9.5 62.1 1 12.5 63.7
0 0.0 63.1 0 3.4 63 0 2.2 62.7 0 6.7 63.1
0 1.1 62.4 0 1.8 64.4 0 0.0 64.2 1 3.6 64.2
0 1.6 63 0 6.2 62.2 0 14.75 62.3 0 .35 63.7
1 13.75 63.8 0 0.0 63.1 1 7.5 62.7 0 1.0 62.9
0 0.0 62.5 1 14.8 61.7 1 3.5 61.6 0 0.0 61.6
0 0.0 61.4 0 .25 61.4 0 1.55 62 1 0.0 61.8
0 0.0 60.9 0 5.9 60.8 0 16.45 60.6 0 2.65 62.9
1 12.5 62.6 0 0.0 62.1 0 14.55 61.7 1 11 61
1 6.75 62.7 0 0.0 62.7 1 0.0 61.7 0 1.75 60.9
0 2.4 60.6 0 10.05 60.4 1 12.75 61.7 0 0.0 61.9
0 5 61.3 0 .6 60.7 0 0.0 60.8 0 .85 60.5
0 .9 59.7 0 .0 59.5 1 8.75 59.6 0 .8 59.1
1 6.6 59.4 0 1.0 58.5 0 0.0 60 1 8.15 59.8
0 0.0 59.7 1 5 59.4 0 2.55 59.2 0 1.2 58.6
0 0.0 60.8 1 11.25 60.4 0 0.0 60.2 0 2 60
0 1.9 59.4 0 .45 59.8 1 0.0 59.7 0 0.0 59.0
1 6.9 59.0 0 2.35 58.6 0 3.95 59.7 0 .6 59.6
1 15 59.4 0 0.0 59.4 0 .95 59.4 0 0.0 59.3
0 1.4 54.2 0 .5 54.0 0 .6 53.8 0 0.0 53.7
0 2.45 53.7 0 1.75 53.1 0 0.0 54.4 0 3.1 54.2
0 10.05 53.9 0 .55 53.2 0 .85 53.2 0 1.1 54.9
0 0.0 54.9 0 0.0 54.5 0 1.45 54.2 0 2.05 54.2
1 10.5 54 0 .5 55.8 1 9.2 55.5 0 .55 55.6
0 0.0 55.5 0 0.96 54.9 0 1 54.6 0 0.0 56.9
0 5.25 56.4 1 0 55.9 0 9 55.8 0 1.6 55.6
1 10.9 57.6 0 0.0 57.7 0 0 57.6 0 2.25 57.8

```

```

0 2.65 57.8 0 .55 58.4 0 0 58.2 1 4.5 58
0 15 58.1 0 0 57.9 0 0 57.3 0 4.2 58.3
0 .55 58.1 1 10 57.9 0 0 57.6 0 7.1 57.3
0 3.2 57.1 1 0 58.9 1 6.8 58.6 0 0 58.7
0 0 57.5 0 2.35 57.2 0 24.9 58 0 2.65 57.9
1 3.7 57.2 0 17.1 57.3 0 0 57.5 0 .95 57.2
0 10.05 53.1 0 1.15 53 1 18.25 53 0 10 52.9
0 .75 52.6 0 0 53.1 0 4.2 53 0 .8 52.9
0 .55 52.7 0 .95 52.6 0 0 52.1 0 3.1 54.1
0 .8 53.7 0 1.55 53.1 0 .4 53.3 0 6.2 53
0 .6 53 0 .4 53.9 1 7.5 53.7 0 7.15 53.4
0 .25 53.2 0 3.6 53.4 0 .95 53.2 0 2.8 54.9
1 20.25 54.9 0 .95 54.6 0 4.25 54.1 0 4.15 54.2
0 10 57.4 0 3.4 57.3 0 0.0 57.3 0 3.6 56.7
0 .9 56.5 0 0.0 56.8 0 0 56.6 1 6.4 56.5
0 .95 56.3 0 1.06 56.3 0 13.3 56.2 0 1.1 56.6
0 17.2 55.9 0 1.65 56 1 5 55.8 0 2.1 55.7
0 .6 57 1 8.25 56.7 0 .9 56.4 0 0.0 56.5
1 12.3 55.2 0 1.15 56.9 0 2.2 56.7 0 3.6 56
1 10 55.5 0 0.6 55.3 0 9.5 56.5 0 .7 56.3
1 9 56.1 0 0 55.9 0 .5 55.5 0 .9 55.4
;

```

```

proc kde data=bronchitis;
  bivar cig poll/plot=all;
run;

```

```

proc corr data=bronchitis plots=matrix;
  var cig poll;
run;

```

```

/*Fitting the Logistic GLM*/
proc logistic data=bronchitis;
  model bronchitis=cig poll/ link=logit
    scale=none
    clparm=wald
    clodds=pl
    rsquare
    outroc=roc;

```

```

run;

proc print data=pred_bronchitis;
run;

    symbol1 i=join v=none c=blue;
proc gplot data=roc;
    plot _sensit*_lmspec_=1/ vaxis= 0 to 1 by 0.1 ;
run;

/*Fitting the Logistic GAM*/
proc gam data=bronchitis plots(clm);
    model bronchitis = spline(cig,df=3) spline(poll,df=3)
/dist=binomial r outroc=roc;
output out=pred_bronshitis all;
run;
proc print data=pred_bronshitis;

data bphat;
    infile 'pred_bronshitis.dat'
    input Obs bronchitis cig poll P_bronchitis P_cig P_poll @@;
    phat=exp(P_bronchitis)/(1+exp(P_bronchitis));
proc print data=bphat (keep=Obs bronchitis cig poll P_bronchitis
phat);
run;

symbol1 i=join v=none c=blue;
proc gplot data=roc;
plot _sensit*_lmspec_=1/ vaxis= 0 to 1 by 0.1 ;
run;

ods graphics off;
ods html close;

```