Lecture 7: Splines and Generalized Additive Models

Computational Statistics

Thierry Denœux

April, 2016

Overview

Introduction

Simple approaches
- Polynomials
- Step functions

Splines
- Regression splines
- Natural splines
- Splines for classification

Smoothing splines
- Definition
- Computation
- Nonparametric logistic regression

Generalized Additive Models
- Principle
- Fitting GAMs

Moving beyond linearity

- Linear models are widely used in econometrics.
- In particular, linear regression, linear discriminant analysis, logistic regression all rely on a linear model.
- It is extremely unlikely that the true function $f(X)$ is actually linear in X. In regression problems, $f(X) = \mathbb{E}(Y|X)$ will typically be nonlinear and nonadditive in X, and representing $f(X)$ by a linear model is usually a convenient, and sometimes a necessary, approximation.
  - Convenient because a linear model is easy to interpret, and is the first-order Taylor approximation to $f(X)$.
  - Sometimes necessary, because with N small and/or p large, a linear model might be all we are able to fit to the data without overfitting.
- Likewise in classification, it is usually assumed that some monotone transformation of $\mathbb{P}(Y = 1|X)$ is linear in X. This is inevitably an approximation.

Linear basis expansion

- The core idea in this chapter is to augment/replace the vector of inputs X with additional variables, which are transformations of X, and then use linear models in this new space of derived input features.
- Denote by $h_m(X) : \mathbb{R}^p \to \mathbb{R}$ the m-th transformation of X, $m = 1, ..., M$. We then model

$$f(X) = \sum_{m=1}^{M} \beta_m h_m(X)$$

  a linear basis expansion in X.
- The beauty of this approach is that once the basis functions h_m have been determined, the models are linear in these new variables, and the fitting proceeds as for linear models.

Popular choices for basis functions h_m

Some simple and widely used examples of the h_m are the following:

- $h_m(X) = X_m,\ m = 1, \ldots, p$ recovers the original linear model.
- $h_m(X) = X_j^2$ or $h_m(X) = X_j X_k$ allows us to augment the inputs with polynomial terms to achieve higher-order Taylor expansions. Note, however, that the number of variables grows exponentially in the degree of the polynomial. A full quadratic model in p variables requires $O(p^2)$ square and cross-product terms, or more generally $O(p^d)$ for a degree-d polynomial.
- $h_m(X) = \log(X_j),\ \sqrt{X_j},\ \ldots$ permits other nonlinear transformations of single inputs. More generally one can use similar functions involving several inputs, such as $h_m(X) = \|X\|$.
- $h_m(X) = I(L_m \le X_k < U_m)$, an indicator for a region of X_k. By breaking the range of X_k up into M_k such nonoverlapping regions results in a model with a piecewise constant contribution for X_k.

Discussion

- Sometimes the problem at hand will call for particular basis functions h_m, such as logarithms or power functions.
- More often, however, we use the basis expansions as a device to achieve more flexible representations for $f(X)$.
- Polynomials are an example of the latter, although they are limited by their global nature – tweaking the coefficients to achieve a functional form in one region can cause the function to flap about madly in remote regions.

Overview

Introduction

Simple approaches
- Polynomials
- Step functions

Splines
- Regression splines
- Natural splines
- Splines for classification

Smoothing splines
- Definition
- Computation
- Nonparametric logistic regression

Generalized Additive Models
- Principle
- Fitting GAMs

Overview

Introduction

Simple approaches
- Polynomials
- Step functions

Splines
- Regression splines
- Natural splines
- Splines for classification

Smoothing splines
- Definition
- Computation
- Nonparametric logistic regression

Generalized Additive Models
- Principle
- Fitting GAMs

Fitting polynomials

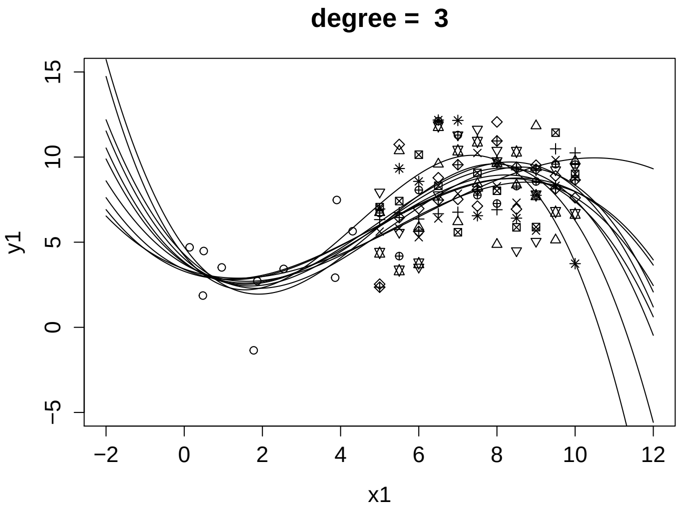
- In most of this lecture, we assume $p = 1$.
- Create new variables $h_1(X) = X$, $h_2(X) = X^2$, $h_3(X) = X^3$, etc. and then do multiple linear regression on the transformed variables.
- We either fix the degree d at some reasonably low value, else use cross-validation to choose d.
- Polynomials have unpredictable tail behavior – very bad for extrapolation.

Example in R

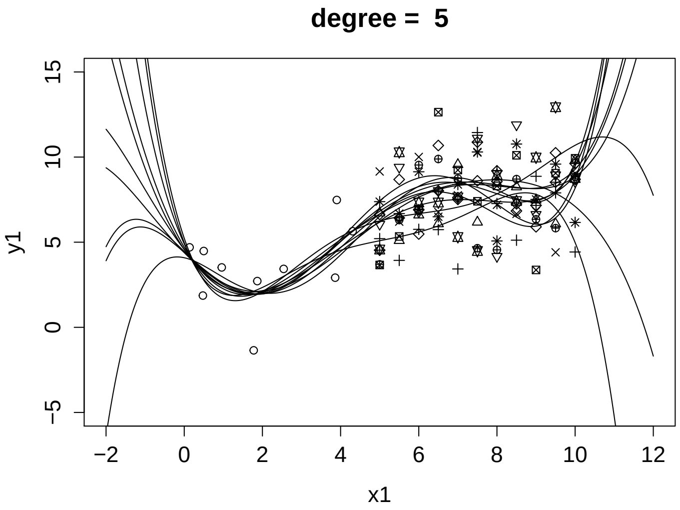
```
x=seq(0,10,0.5)
n<-length(x)
y1=x[1:10]+2*cos(x[1:10])+2*rnorm(10)
xtest<- seq(-2,12,0.01)
ftest<- xtest+2*cos(xtest)
d<-3

plot(x1,y1,xlim=c(-2,12),ylim=c(-5,15),
                 main=paste('degree = ',as.character(d)))
for(i in 1:10){
    y2=x[11:21]+2*cos(x[11:21])+2*rnorm(11)
    points(x[11:21],y2,pch=i+1)
    y<-c(y1,y2)
    reg<-lm(y ~ poly(x,degree=d))
    ypred<-predict(reg,newdata=data.frame(x=xtest),interval="c")
    lines(xtest,ypred[,"fit"],lty=1)
}
```

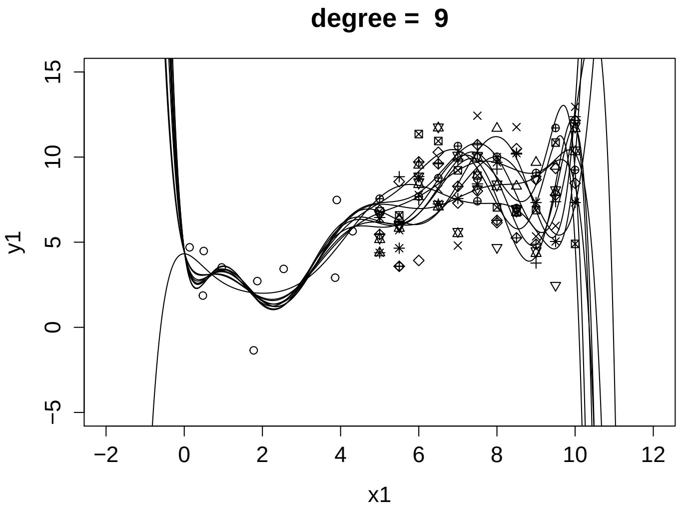
Result, $d = 2$



Result, $d = 3$



Result, $d = 4$



Overview

Introduction

Simple approaches
- Polynomials
- Step functions

Splines
- Regression splines
- Natural splines
- Splines for classification

Smoothing splines
- Definition
- Computation
- Nonparametric logistic regression

Generalized Additive Models
- Principle
- Fitting GAMs

Step Functions

- Another way of creating transformations of a variable is to cut the variable into distinct regions.

$$h_1(X) = I(X < \xi_1), h_2(X) = I(\xi_1 \leq X < \xi_2), \ldots, \\ h_M(X) = I(X \geq \xi_{M-1})$$

- Since the basis functions are positive over disjoint regions, the least squares estimate of the model $f(X) = \sum_{m=1}^{M} \beta_m h_m(X)$ is $\widehat{\beta}_m = \overline{Y}_m$, the mean of Y in the m-th region.

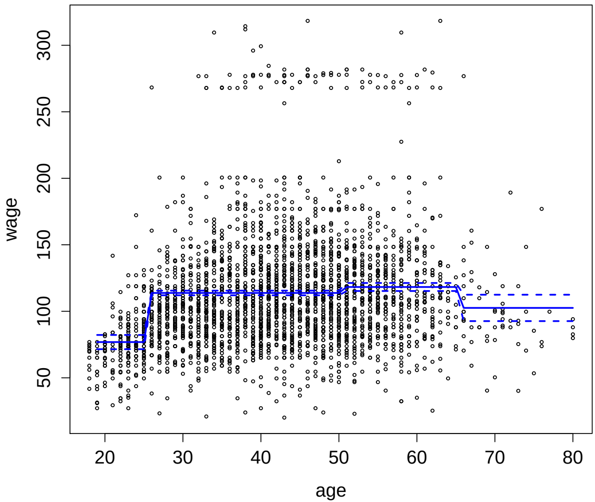
Example in R

```
library("ISLR")

reg<-lm(wage ~ cut(age, c(18, 25, 50, 65, 90)),data=Wage)
ypred<-predict(reg,newdata=data.frame(age=18:80),interval="c")

plot(Wage$age,Wage$wage,cex=0.5,xlab="age",ylab="wage")
lines(18:80,ypred[,"fit"],lty=1,col="blue",lwd=2)
lines(18:80,ypred[,"lwr"],lty=2,col="blue",lwd=2)
lines(18:80,ypred[,"upr"],lty=2,col="blue",lwd=2)
```

Result



Step functions – continued

- Easy to work with. Creates a series of dummy variables representing each group.
- Useful way of creating interactions that are easy to interpret. For example, interaction effect of Year and Age:

$$I(Year < 2005) \cdot Age, I(Year \geq 2005) \cdot Age$$

  would allow for different linear functions in each age category.
- Choice of cutpoints or knots can be problematic. For creating nonlinearities, smoother alternatives such as splines are available.

Overview

Introduction

Simple approaches
- Polynomials
- Step functions

Splines
- Regression splines
- Natural splines
- Splines for classification

Smoothing splines
- Definition
- Computation
- Nonparametric logistic regression

Generalized Additive Models
- Principle
- Fitting GAMs

Overview

Introduction

Simple approaches
- Polynomials
- Step functions

Splines
- Regression splines
- Natural splines
- Splines for classification

Smoothing splines
- Definition
- Computation
- Nonparametric logistic regression

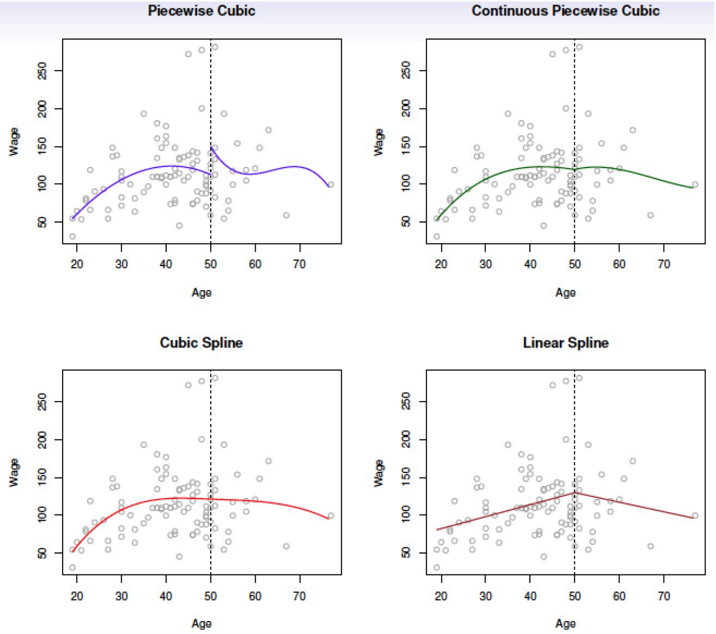
Generalized Additive Models
- Principle
- Fitting GAMs

Piecewise Polynomials

- Instead of a single polynomial in X over its whole domain, we can rather use different polynomials in regions defined by knots. E.g. (see figure)

$$y_i = \begin{cases} \beta_{01} + \beta_{11}x_i + \beta_{21}x_i^2 + \beta_{31}x_i^3 + \epsilon_i & \text{if } x_i < \xi, \\ \beta_{02} + \beta_{12}x_i + \beta_{22}x_i^2 + \beta_{32}x_i^3 + \epsilon_i & \text{if } x_i \geq \xi, \end{cases}$$

- Better to add constraints to the polynomials, e.g. continuity.
- Splines have the "maximum" amount of continuity.


Piecewise Cubic
Continuous Piecewise Cubic
Cubic Spline
Linear Spline
Wage
Age
50
100
150
200
250
20
30
40
50
60
70

Linear Splines

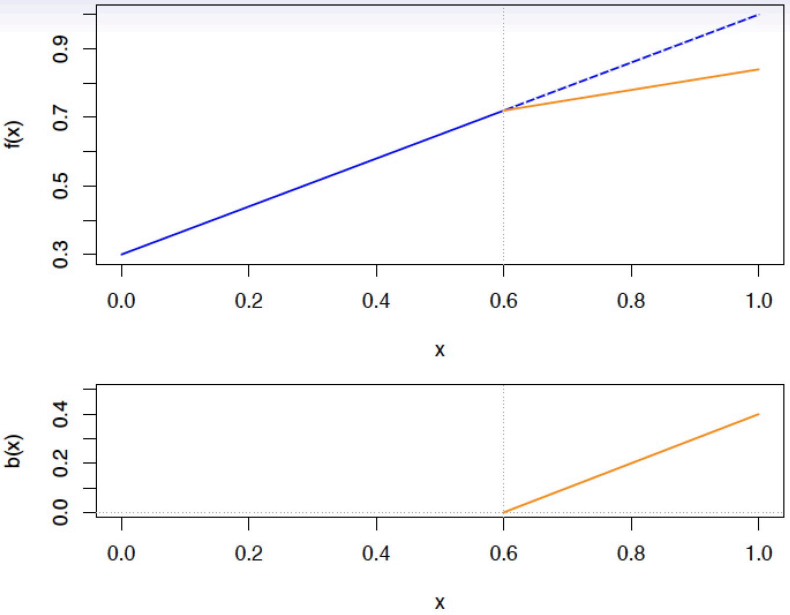
- A linear spline with knots at ξ_k, $k = 1, \dots, K$ is a piecewise linear polynomial continuous at each knot.
- The set of linear splines with fixed knots is a vector space.
- The number of degrees of freedom is $2(K+1) - K = K + 2$. We can thus decompose linear splines on a basis of $K + 2$ basis functions,

$$y = \sum_{m=1}^{K+2} \beta_m h_m(x) + \epsilon.$$

- The basis functions can be chosen as

$$\begin{aligned} h_1(x) &= 1 \\ h_2(x) &= x \\ h_{k+2}(x) &= (x - \xi_k)_+, \quad k = 1, \dots, K, \end{aligned}$$

where $(\cdot)_+$ denotes the positive part, i.e., $(x - \xi_k)_+ = x - \xi_k$ if $x > \xi_k$ and $(x - \xi_k)_+ = 0$ otherwise.


f(x)
0.3
0.5
0.7
0.9
0.0
0.2
0.4
0.6
0.8
1.0
x
b(x)
0.0
0.2
0.4
0.0
0.2
0.4
0.6
0.8
1.0
x

Cubic Splines

- A cubic spline with knots at ξ_k, $k = 1, \ldots, K$ is a piecewise cubic polynomial with continuous derivatives up to order 2 at each knot.
- Enforcing one more order of continuity would lead to a global cubic polynomial.
- Again, the set of cubic splines with fixed knots is a vector space, and the number of degrees of freedom is $4(K+1) - 3K = K + 4$. We can thus decompose cubic splines on a basis of $K + 4$ basis functions,

$$y = \sum_{m=1}^{K+4} \beta_m h_m(x) + \epsilon.$$

- We can choose truncated power basis functions,

$$h_k(x) = x^{k-1}, \quad k = 1, \ldots, 4,$$
$$h_{k+4}(x) = (x - \xi_k)_+^3, \quad k = 1, \ldots, K.$$

order-M splines

- More generally, an order-M spline with knots ξ_k, $k = 1, \ldots, K$ is a piecewise-polynomial of order $M - 1$, which has continuous derivatives up to order $M - 2$.
- A cubic spline has $M = 4$. A piecewise-constant function is an order-1 spline, while a continuous piecewise linear function is an order-2 spline.
- The general form for the truncated-power basis set is

$$
\begin{aligned}
h_k(x) &= x^{k-1}, \quad k = 1, \ldots, M, \\
h_{k+M}(x) &= (x - \xi_k)_+^{M-1}, \quad k = 1, \ldots, K.
\end{aligned}
$$

- It is claimed that cubic splines are the lowest-order spline for which the knot-discontinuity is not visible to the human eye. There is seldom any good reason to go beyond cubic-splines.
- In practice the most widely used orders are $M = 1, 2$ and 4.

Splines in R

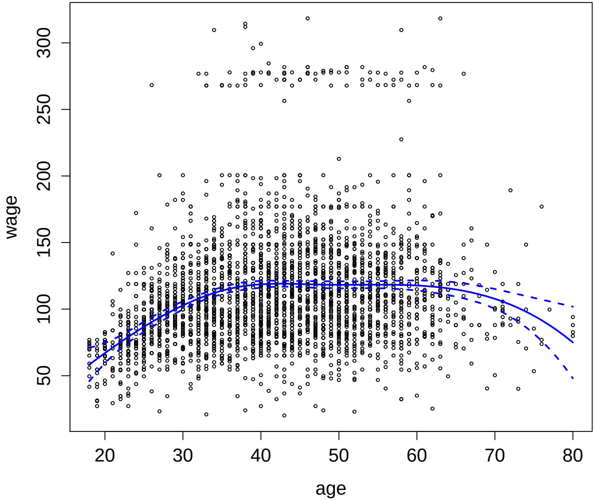
```
library('splines')
fit<-lm(wage~bs(age,5),data=Wage)

ypred<-predict(fit,newdata=data.frame(age=18:80),interval="c")

plot(Wage$age,Wage$wage,cex=0.5,xlab="age",ylab="wage")
lines(18:80,ypred[,"fit"],lty=1,col="blue",lwd=2)
lines(18:80,ypred[,"lwr"],lty=2,col="blue",lwd=2)
lines(18:80,ypred[,"upr"],lty=2,col="blue",lwd=2)
```

- By default, `degree=3`, and the intersect is not included in the basis functions.
- The number of knots is `df-degree`. If not specified, the knots are placed at quantiles.

Result



B-spline basis

- Since the space of spline functions of a particular order and knot sequence is a vector space, there are many equivalent bases for representing them (just as there are for ordinary polynomials.)
- While the truncated power basis is conceptually simple, it is not too attractive numerically: powers of large numbers can lead to severe rounding problems.
- In practice, we often use another basis: the B-spline basis, which allows for efficient computations even when the number of knots K is large (each basis function has a local support).

B-spline basis

Construction

- Before we can get started, we need to augment the knot sequence.
- Let $\xi_0 < \xi_1$ and $\xi_K < \xi_{K+1}$ be two boundary knots, which typically define the domain over which we wish to evaluate our spline. We now define the augmented knot sequence τ such that
    - $\tau_1 \leq \tau_2 \leq \ldots \leq \tau_M \leq \xi_0$
    - $\tau_{j+M} = \xi_j$, $j = 1, \ldots, K$
    - $\xi_{K+1} \leq \tau_{K+M+1} \leq \tau_{K+M+2} \leq \ldots \leq \tau_{K+2M}$.
- The actual values of these additional knots beyond the boundary are arbitrary, and it is customary to make them all the same and equal to ξ_0 and ξ_{K+1}, respectively.

B-spline basis

Construction – Continued

- Denote by $B_{i,m}(x)$ the ith B-spline basis function of order m for the knot-sequence τ, $m \leq M$. They are defined recursively in terms of divided differences as follows:

$$B_{i,1}(x) = \begin{cases} 1 & \text{if } \tau_i \leq x < \tau_{i+1} \\ 0 & \text{otherwise} \end{cases}$$

for $i = 1, \ldots, K + 2M - 1$. (By convention, $B_{i,1} = 0$ if $\tau_i = \tau_{i+1}$).

$$B_{i,m} = \frac{x - \tau_i}{\tau_{i+m-1} - \tau_i} B_{i,m-1}(x) + \frac{\tau_{i+m} - x}{\tau_{i+m} - \tau_{i+1}} B_{i+1,m-1}(x)$$

for $i = 1, ..., K + 2M - m$.

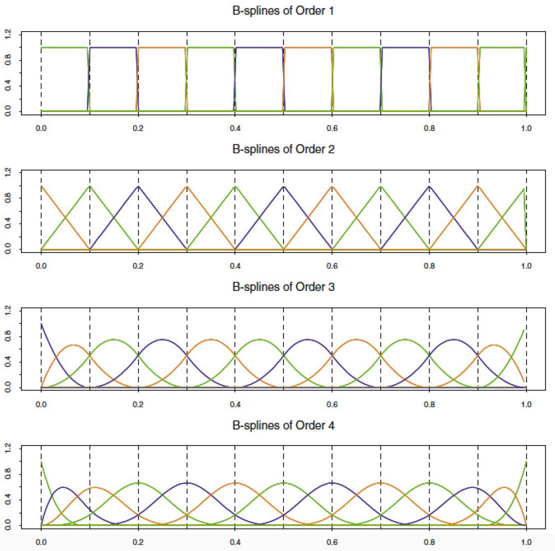
- Thus with $M = 4$, $B_{i,4}, i = 1, \ldots, K + 4$ are the $K + 4$ cubic B-spline basis functions for the knot sequence ξ.

B-spline basis

Properties

- The B-splines span the space of cubic splines for the knot sequence ξ.
- They have local support and they are nonzero on an interval spanned by $M + 1$ knots (see next slide).

Sequence of B-splines up to order 4 with 10 knots evenly spaced from 0 to 1



Overview

Introduction

Simple approaches
- Polynomials
- Step functions

Splines
- Regression splines
- Natural splines
- Splines for classification

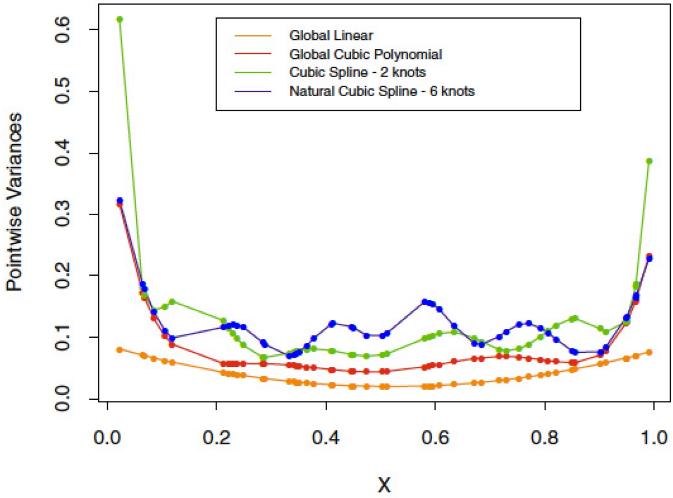
Smoothing splines
- Definition
- Computation
- Nonparametric logistic regression

Generalized Additive Models
- Principle
- Fitting GAMs

Variance of splines beyond the boundary knots

- We know that the behavior of polynomials fit to data tends to be erratic near the boundaries, and extrapolation can be dangerous.
- These problems are exacerbated with splines. The polynomials fit beyond the boundary knots behave even more wildly than the corresponding global polynomials in that region.

Example


Global Linear
Global Cubic Polynomial
Cubic Spline - 2 knots
Natural Cubic Spline - 6 knots
Pointwise Variances
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.0
0.2
0.4
0.6
0.8
1.0
X

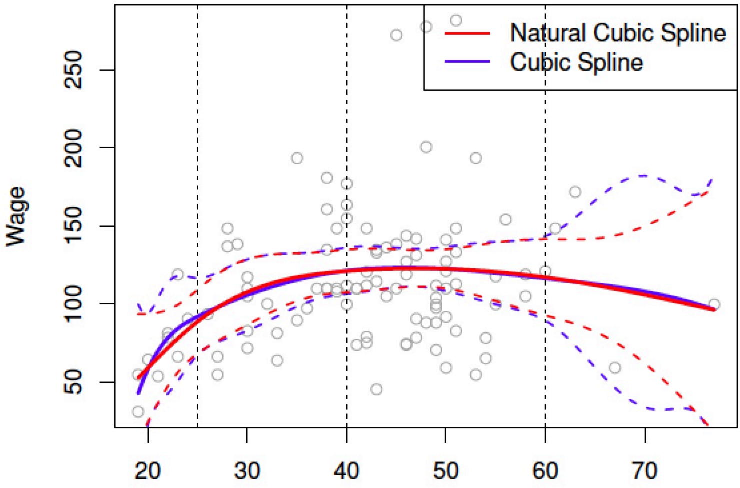
Explanation of the previous figure

- Pointwise variance curves for four different models, with X consisting of 50 points drawn at random from $U[0, 1]$, and an assumed error model with constant variance.
- The linear and cubic polynomial fits have 2 and 4 df, respectively, while the cubic spline and natural cubic spline each have 6 df.
- The cubic spline has two knots at 0.33 and 0.66, while the natural spline has boundary knots at 0.1 and 0.9, and four interior knots uniformly spaced between them.

Natural cubic spline

- A natural cubic spline adds additional constraints, namely that the function is linear beyond the boundary knots.
- This frees up four degrees of freedom (two constraints each in both boundary regions), which can be spent more profitably by putting more knots in the interior region.
- There will be a price paid in bias near the boundaries, but assuming the function is linear near the boundaries (where we have less information anyway) is often considered reasonable.

Example


Natural Cubic Spline
Cubic Spline
Wage
250
200
150
100
50
20
30
40
50
60
70

Natural cubic spline basis

- A natural cubic spline with K knots has K degrees of freedom: it can ve represented by K basis functions.
- One can start from a basis for cubic splines, and derive the reduced basis by imposing the boundary constraints. For example, starting from the truncated power series basis,

$$f(X) = \sum_{j=0}^{3} \beta_j X^j + \sum_{k=1}^{K} \theta_k (X - \xi_k)_+^3,$$

the contraints $f''(X) = 0$ and $f^{(3)}(X) = 0$ for $X < \xi_1$ and $X > \xi_K$ lead to the conditions

$$\beta_2 = \beta_3 = 0, \quad \sum_{k=1}^{K} \theta_k = 0, \quad \sum_{k=1}^{K} \xi_k \theta_k = 0$$

Natural cubic spline basis – continued

- These conditions are automatically satisfied by choosing the following basis,

$$N_1(X) = 1, \quad N_2(X) = X,$$

$$N_{k+2}(X) = d_k(X) - d_{K-1}(X), \quad k = 1, \dots, K-2$$

with

$$d_k = \frac{(X - \xi_k)_+^3 - (X - \xi_K)_+^3}{\xi_K - \xi_k}$$

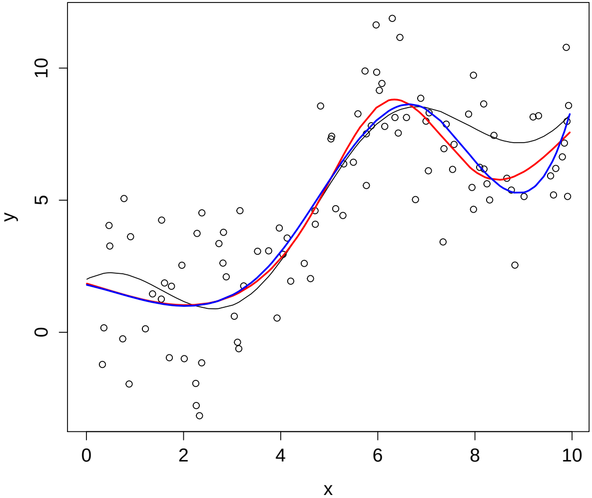
Example in R

```
fit1<-lm(y ~ ns(x,df=5))
fit2<-lm(y ~ bs(x,df=5))

ypred1<-predict(fit1,newdata=data.frame(x=xtest),interval="c")
ypred2<-predict(fit2,newdata=data.frame(x=xtest),interval="c")

plot(x,y,xlim=range(xtest))
lines(xtest,ftest)
lines(xtest,ypred1[,"fit"],lty=1,col="red",lwd=2)
lines(xtest,ypred2[,"fit"],lty=1,col="blue",lwd=2)
```

Result



Overview

Introduction

Simple approaches
- Polynomials
- Step functions

Splines
- Regression splines
- Natural splines
- Splines for classification

Smoothing splines
- Definition
- Computation
- Nonparametric logistic regression

Generalized Additive Models
- Principle
- Fitting GAMs

Using splines with logistic regression

- Until now, we have discussed regression problems. However, splines can also be used when the response variable is qualitative.
- Consider, for instance, natural splines with K knots. For binary classification, we can fit the logistic regression model,

$$\log \frac{\mathbb{P}(Y = 1 \; X = x)}{\mathbb{P}(Y = 0 \; X = x)} = f(x)$$

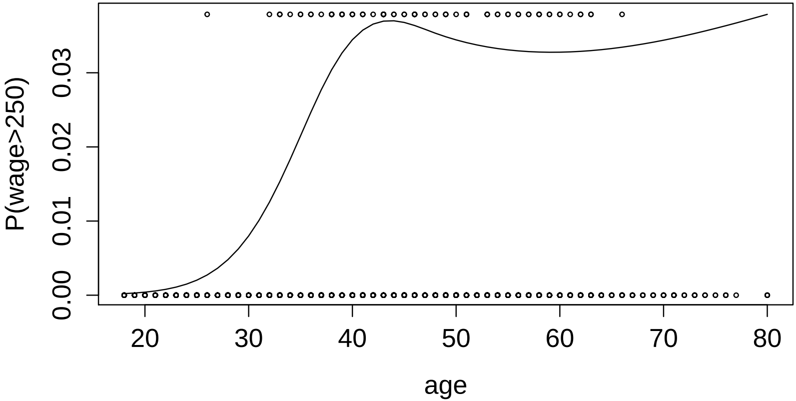
  with $f(x) = \sum_{k=1}^{K} \beta_k N_k(x)$.
- Once the basis functions have been defined, we just need to estimate coefficients β_k using a standard logistic regression procedure.
- A smooth estimate of the conditional probability $\mathbb{P}(Y = 1|x)$ can then be used for classification or risk scoring.

Example in R

```
class<-glm(I(wage>250) ~ ns(age,3),data=Wage,family='binomial')
proba<-predict(class,newdata=data.frame(age=18:80),type='response')

plot(18:80,proba,xlab="age",ylab="P(wage>250)",type="l")
ii<-which(Wage$wage>250)
points(Wage$age[ii],rep(max(proba),length(ii)),cex=0.5)
points(Wage$age[-ii],rep(0,nrow(Wage)-length(ii)),cex=0.5)
```

Result



Overview

Introduction

Simple approaches
- Polynomials
- Step functions

Splines
- Regression splines
- Natural splines
- Splines for classification

Smoothing splines
- Definition
- Computation
- Nonparametric logistic regression

Generalized Additive Models
- Principle
- Fitting GAMs

Overview

Introduction

Simple approaches
- Polynomials
- Step functions

Splines
- Regression splines
- Natural splines
- Splines for classification

Smoothing splines
- Definition
- Computation
- Nonparametric logistic regression

Generalized Additive Models
- Principle
- Fitting GAMs

Smoothing splines

Problem formulation

- Here we discuss a spline basis method that avoids the knot selection problem completely by using a maximal set of knots. The complexity of the fit is controlled by regularization.
- Problem: among all functions $f(x)$ with two continuous derivatives, find one that minimizes the penalized residual sum of squares

$$RSS(f, \lambda) = \sum_{i=1}^{N} (y_i - f(x_i))^2 + \lambda \int [f''(t)]^2 dt,$$

  where λ is a fixed smoothing parameter.
- The first term measures closeness to the data, while the second term penalizes curvature in the function, and λ establishes a tradeoff between the two. Special cases: $\lambda = 0$ (no constraint on f) and $\lambda = \infty$ (f has to be linear).

Smoothing splines
Solution

- It can be shown that this problem has an explicit, finite-dimensional, unique minimizer which is a natural cubic spline with knots at the unique values of the $x_i, i = 1, \dots, N$.
- At face value it seems that the family is still over-parametrized, since there are as many as N knots, which implies N degrees of freedom. However, the penalty term translates to a penalty on the spline coefficients, which are shrunk some of the way toward the linear fit.
- The solution is thus of the form

$$f(x) = \sum_{j=1}^{N} N_j(x)\theta_j,$$

where the $N_j(x)$ are an N-dimensional set of basis functions for representing this family of natural splines.

Overview

Introduction

Simple approaches
- Polynomials
- Step functions

Splines
- Regression splines
- Natural splines
- Splines for classification

Smoothing splines
- Definition
- Computation
- Nonparametric logistic regression

Generalized Additive Models
- Principle
- Fitting GAMs

Computation

- The criterion can be written as

$$RSS(\theta, \lambda) = (\boldsymbol{y} - \boldsymbol{N}\theta)^T(\boldsymbol{y} - \boldsymbol{N}\theta) + \lambda\theta^T\boldsymbol{\Omega}_N\theta,$$

  where $\{\boldsymbol{N}\}_{ij} = N_j(x_i)$ and $\{\boldsymbol{\Omega}_N\}_{jk} = \int N_j''(t)N_k''(t)dt$.
- The solution is

$$\widehat{\theta} = (\boldsymbol{N}^T\boldsymbol{N} + \lambda\boldsymbol{\Omega}_N)^{-1}\boldsymbol{N}^T\boldsymbol{y},$$

  a generalized ridge regression.
- The fitted smoothing spline is given by

$$\widehat{f}(x) = \sum_{j=1}^{N} N_j(x)\widehat{\theta}_j.$$

- In practice, when N is large, we can use only a subset of the N interior knots (rule of thumb: number of knots proportional to $\log N$).

Degrees of freedom

- Denote by $\widehat{\boldsymbol{f}}$ the N-vector of fitted values $f(x_i)$ at the training predictors x_i. Then,

$$\widehat{\boldsymbol{f}} = \boldsymbol{N}\widehat{\theta} = (\boldsymbol{N}^T\boldsymbol{N} + \lambda\boldsymbol{\Omega}_N)^{-1}\boldsymbol{N}^T\boldsymbol{y} = \boldsymbol{S}_\lambda\boldsymbol{y}$$

- As matrix $\boldsymbol{S}_\lambda$ does not depend on $\boldsymbol{y}$, the smoothing spline is a linear smoother.
- In the case of cubic spline with knot sequence ξ and, we have

$$\widehat{\boldsymbol{f}} = \boldsymbol{B}_\xi\widehat{\theta} = (\boldsymbol{B}_\xi^T\boldsymbol{B}_\xi)^{-1}\boldsymbol{B}_\xi^T\boldsymbol{y} = \boldsymbol{H}_\xi\boldsymbol{y},$$

  where $\boldsymbol{B}_\xi$ is the $N \times M$ matrix of basis functions. The degrees of freedom is $M = \mathrm{trace}(\boldsymbol{H}_\xi)$.
- By analogy, the effective degrees of freedom of a smoothing spline is defined as

$$\mathrm{df}_\lambda = \mathrm{trace}(\boldsymbol{S}_\lambda)$$

Selection of smoothing parameters

- As $\lambda \to 0$, $\mathrm{df}_\lambda \to N$ and $\boldsymbol{S}_\lambda \to \boldsymbol{I}$. As $\lambda \to \infty$, $\mathrm{df}_\lambda \to 2$ and $\boldsymbol{S}_\lambda \to \boldsymbol{H}$, the hat matrix for linear regression on $\boldsymbol{x}$.
- Since df_λ is monotone in λ, we can invert the relationship and specify λ by fixing df_λ (this can be achieved by simple numerical methods). Using df in this way provides a uniform approach to compare many different smoothing methods.
- The leave-one-out (LOO) cross-validated error is given by

$$RSS_{cv}(\lambda) = \sum_{i=1}^{N}(y_i - \widehat{f}_\lambda^{(-i)}(x_i))^2 = \sum_{i=1}^{N}\left[\frac{y_i - \widehat{f}_\lambda(x_i)}{1 - \{\boldsymbol{S}_\lambda\}_{ii}}\right]^2$$

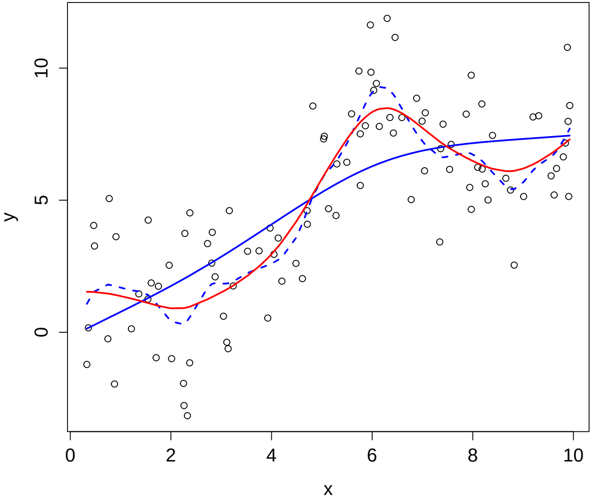
Smoothing splines in R

```
ss1<-smooth.spline(x,y,df=3)
ss2<-smooth.spline(x,y,df=15)
ss<-smooth.spline(x,y)

plot(x,y)
lines(x,ss1$y,col="blue",lwd=2)
lines(x,ss2$y,col="blue",lwd=2,lty=2)
lines(x,ss$y,col="red",lwd=2)

> ss$df
7.459728
```

Result



Overview

Introduction

Simple approaches
- Polynomials
- Step functions

Splines
- Regression splines
- Natural splines
- Splines for classification

Smoothing splines
- Definition
- Computation
- Nonparametric logistic regression

Generalized Additive Models
- Principle
- Fitting GAMs

Application to logistic regression

- The smoothing spline problem has been posed in a regression setting. It is typically straightforward to transfer this technology to other domains.
- Here we consider logistic regression with a single quantitative input X. The model is

$$\log \frac{\mathbb{P}(Y = 1|X = x)}{\mathbb{P}(Y = 0\ X = x)} = f(x),$$

which implies

$$\mathbb{P}(Y = 1|X = x) = \frac{e^{f(x)}}{1 + e^{f(x)}} = p(x).$$

Penalized log-likelihood

- We construct the penalized log-likelihood criterion

$$\begin{aligned}\ell(f;\lambda) &= \sum_{i=1}^{N}[y_i \log p(x_i) + (1-y_i)\log(1-p(x_i))] - \frac{1}{2}\lambda\int\{f''(t)\}^2 dt \\ &= \sum_{i=1}^{N}[y_i f(x_i) - \log(1+e^{f(x)})] - \frac{1}{2}\lambda\int\{f''(t)\}^2 dt\end{aligned}$$

- As before, the optimal f is a finite-dimensional natural spline with knots at the unique values of x. We can represent f as

$$f(x) = \sum_{j=1}^{N} N_j(x)\theta_j.$$

Optimization

- We compute the first and second derivatives

$$\frac{\partial \ell(\theta)}{\partial \theta} = \boldsymbol{N}^T(\boldsymbol{y} - \boldsymbol{p}) - \lambda \boldsymbol{\Omega} \theta$$

$$\frac{\partial^2 \ell(\theta)}{\partial \theta \partial \theta^T} = -\boldsymbol{N}^T \boldsymbol{W} \boldsymbol{N} - \lambda \boldsymbol{\Omega},$$

  where $\boldsymbol{p}$ is the N-vector with elements $p(xi)$, and $\boldsymbol{W}$ is a diagonal matrix of weights $p(x_i)(1 - p(x_i))$.

- Parameters θ_j can be estimated using the Newton method,

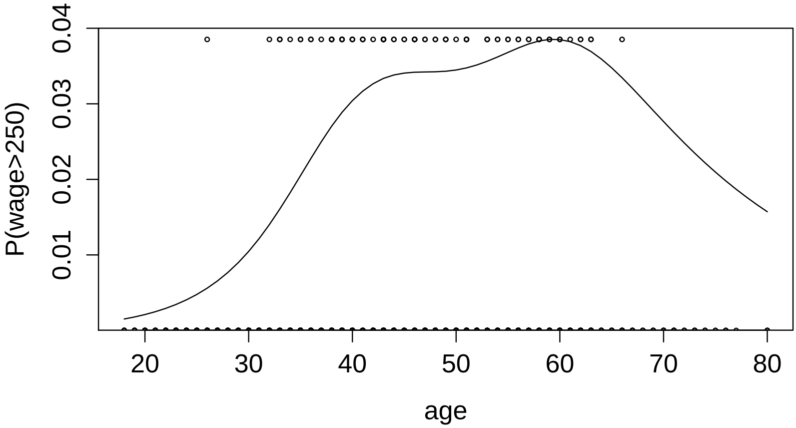
$$\theta^{new} = \theta^{old} - \left(\frac{\partial^2 \ell(\theta^{old})}{\partial \theta \partial \theta^T} \right)^{-1} \frac{\partial \ell(\theta^{old})}{\partial \theta}$$

Nonparametric logistic regression in R

```
library(gam)
class<-gam(I(wage>250) ~ s(age,df=3),data=Wage,family='binomial')
proba<-predict(class,newdata=data.frame(age=18:80),type='response')

plot(18:80,proba,xlab="age",ylab="P(wage>250)",type="l")
ii<-which(Wage$wage>250)
points(Wage$age[ii],rep(max(proba),length(ii)),cex=0.5)
points(Wage$age[-ii],rep(0,nrow(Wage)-length(ii)),cex=0.5)
```

Result



Overview

Introduction

Simple approaches
- Polynomials
- Step functions

Splines
- Regression splines
- Natural splines
- Splines for classification

Smoothing splines
- Definition
- Computation
- Nonparametric logistic regression

Generalized Additive Models
- Principle
- Fitting GAMs

Overview

Introduction

Simple approaches
- Polynomials
- Step functions

Splines
- Regression splines
- Natural splines
- Splines for classification

Smoothing splines
- Definition
- Computation
- Nonparametric logistic regression

Generalized Additive Models
- Principle
- Fitting GAMs

Motivation

- Regression models play an important role in many data analyses, providing prediction and classification rules, and data analytic tools for understanding the importance of different inputs.
- Although attractively simple, the traditional linear model often fails in these situations: in real life, effects are often not linear.
- Here, we describe more automatic flexible statistical methods that may be used to identify and characterize nonlinear regression effects. These methods are called generalized additive models (GAMs).

GAM for regression

- In the regression setting, a generalized additive model has the form

$$\mathbb{E}(Y|X_1, X_2, \ldots, X_p) = \alpha + f_1(X_1) + f_2(X_2) + \ldots + f_p(X_p)$$

- As usual $X_1, X_2, \ldots, X_p$ represent predictors and Y is the outcome.
- The f_j's are unspecified smooth (nonparametric) functions.

GAM for binary classification

- For two-class classification, recall the logistic regression model for binary data discussed previously. We relate the mean of the binary response $\mu(X) = \mathbb{P}(Y = 1|X)$ to the predictors via a linear regression model and the logit link function:

$$\log \frac{\mu(X)}{1-\mu(X)} = \alpha + \beta_1 X_1 + \ldots + \beta_p X_p$$

- The additive logistic regression model replaces each linear term by a more general functional form

$$\log \frac{\mu(X)}{1-\mu(X)} = \alpha + f_1(X_1) + \ldots + f_p(X_p)$$

  where again each f_j is an unspecified smooth function.
- While the nonparametric form for the functions f_j makes the model more flexible, the additivity is retained and allows us to interpret the model in much the same way as before.

GAM: general form

- In general, the conditional mean $\mu(X)$ of a response Y is related to an additive function of the predictors via a link function g:

$$g[\mu(X)] = \alpha + f_1(X_1) + \ldots + f_p(X_p)$$

- Examples of classical link functions are the following:
  - $g(\mu) = \mu$ is the identity link, used for linear and additive models for Gaussian response data.
  - $g(\mu) = \text{logit}(\mu)$ as above, or $g(\mu) = \text{probit}(\mu)$, the probit link function, for modeling binomial probabilities. The probit function is the inverse Gaussian cumulative distribution function: $\text{probit}(\mu) = \Phi^{-1}(\mu)$.
  - $g(\mu) = \log(\mu)$ for log-linear or log-additive models for Poisson count data.

Mixing linear and nonlinear effects, interactions

- We can easily mix in linear and other parametric forms with the nonlinear terms, a necessity when some of the inputs are qualitative variables (factors).
- The nonlinear terms are not restricted to main effects either; we can have nonlinear components in two or more variables, or separate curves in X_j for each level of the factor X_k, e.g.,
    - $g(\mu) = X^T\beta + \sum_k \alpha_k I(V = k) + f(Z)$ – a semiparametric model, where X is a vector of predictors to be modeled linearly, α_k the effect for the kth level of a qualitative input V, and the effect of predictor Z is modeled nonparametrically.
    - $g(\mu) = f(X) + \sum_k g_k(Z)I(V = k)$ – again k indexes the levels of a qualitative input V, and thus creates an interaction term for the effect of V and Z,
    - etc...

Overview

Introduction

Simple approaches
- Polynomials
- Step functions

Splines
- Regression splines
- Natural splines
- Splines for classification

Smoothing splines
- Definition
- Computation
- Nonparametric logistic regression

Generalized Additive Models
- Principle
- Fitting GAMs

GAMs with natural splines

- If we model each function f_j as a natural spline, then we can fit the resulting model using simple least square (regression) or likelihood maximization algorithm (classification).
- For instance, with natural cubic splines, we have the following GAM:

$$g(\mu) = \sum_{j=1}^{p} \sum_{k=1}^{K(j)} \beta_{jk} N_k(X_j) + \epsilon,$$

where $K(j)$ is the number of knots for variable j.

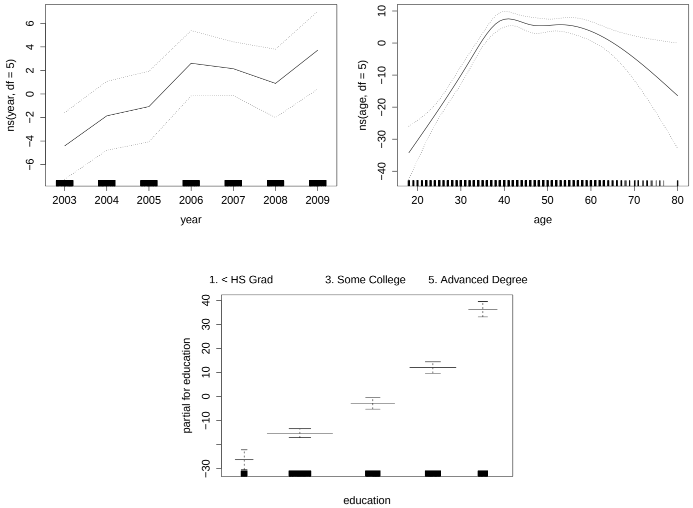
Example in R

```
library("ISLR") # For the Wage data
library("splines")

fit1<-lm(wage ~ ns(year,df=5)+ns(age,df=5)+education,data=Wage)

library("gam")
fit2<-gam(wage ~ ns(year,df=5)+ns(age,df=5)+education,data=Wage)
plot(fit2,se=TRUE)
```

Result


ns(year, df = 5)
6
4
2
0
−2
−4
−6
2003
2004
2005
2006
2007
2008
2009
year
ns(age, df = 5)
10
0
−10
−20
−30
−40
20
30
40
50
60
70
80
age
1. < HS Grad
3. Some College
5. Advanced Degree
partial for education
40
30
20
10
0
−10
−30
education

GAMs with smoothing splines

- Consider an additive model of the form

$$Y = \alpha + f_1(X_1) + f_2(X_2) + \ldots + f_p(X_p) + \epsilon,$$

  where the error term ϵ has mean zero.
- We can specify a penalized sum of squares for this problem,

$$SS(\alpha, f_1, \ldots, f_p) = \sum_{i=1}^{N} \left(y_i - \alpha - \sum_{j=1}^{p} f_j(x_{ij}) \right)^2 + \sum_{j=1}^{p} \lambda_j \int f_j''(t_j)^2 dt_j,$$

  where the $\lambda_j \geq 0$ are tuning parameters.
- It can be shown that the minimizer of SS is an additive cubic spline model; each of the functions f_j is a cubic spline in the component X_j, with knots at each of the unique values of $x_{ij}, i = 1, \ldots, N$.

Unicity of the solution

- Without further restrictions on the model, the solution is not unique.
- The constant α is not identifiable, since we can add or subtract any constants to each of the functions f_j, and adjust α accordingly.
- The standard convention is to assume that $\sum_{i=1}^{N} f_j(x_{ij}) = 0$ for all j – the functions average zero over the data.
- It is easily seen that $\widehat{\alpha} = \text{ave}(y_i)$ in this case.
- If in addition to this restriction, the matrix of input values (having ijth entry xij) has full column rank, then SS is a strictly convex criterion and the minimizer is unique.

Backfitting algorithm

- A simple iterative procedure exists for finding the solution.
- We set $\widehat{\alpha} = \text{ave}(y_i)$, and it never changes.
- We apply a cubic smoothing spline $\mathcal{S}_j$ to the targets $\{y_i - \widehat{\alpha} - \widehat{f}(x_{ik})\}_{i=1}^N$, as a function of x_{ij} to obtain a new estimate $\widehat{f}_j$.
- This is done for each predictor in turn, using the current estimates of the other functions $\widehat{f}_k$ when computing $y - \widehat{\alpha} - \sum_{k \neq j} \widehat{f}_k(x_{ik})$.
- The process is continued until the estimates f_j stabilize.
- This procedure (known as backfitting) is grouped cyclic coordinate descent algorithm.

Backfitting algorithm

1. Initialize: $\widehat{\alpha} = \text{ave}(y_i)$, $\widehat{f}_j = 0$, $\forall i, j$.
2. Cycle: $j = 1, 2, \ldots, p, 1, 2, \ldots, p, \ldots,$

$$\widehat{f}_j \leftarrow \mathcal{S}_j \left[\{ y_i - \widehat{\alpha} - \sum_{k \neq j} \widehat{f}_k(x_{ik}) \}_{i=1}^N \right]$$

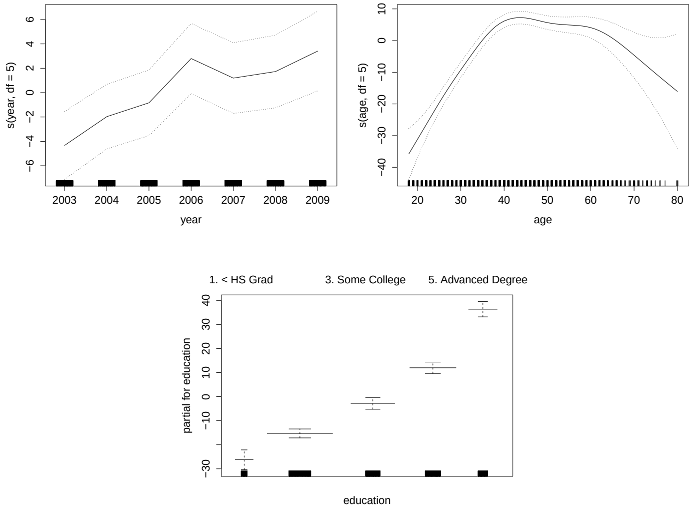
$$\widehat{f}_j \leftarrow \widehat{f}_j - \frac{1}{N} \sum_{i=1}^{N} \widehat{f}_j(x_{ij})$$

until the functions $\widehat{f}_j$ change less than a prespecified threshold.

Example in R

```
library("gam")
fit3<-gam(wage ~ s(year,df=5)+s(age,df=5)+education,data=Wage)
plot(fit3,se=TRUE)
```

Result


s(year, df = 5)
−6
−4
−2
0
2
4
6
2003
2004
2005
2006
2007
2008
2009
year
s(age, df = 5)
−40
−30
−20
−10
0
10
20
30
40
50
60
70
80
age
1. < HS Grad
3. Some College
5. Advanced Degree
partial for education
−30
−10
0
10
20
30
40
education