Moving Beyond Linearity

by

AINDRILA GARAI

MSC STATISTICS, IIT KANPUR

aindrilag22@iitk.ac.in

Polynomial Regression:

The nonlinear relationship between the predictors and the response is defined by a polynomial function

$$y_i = \beta_0 + \beta_1 x_i + \beta_2 x_i^2 + \beta_3 x_i^3 + \cdots + \beta_d x_i^d + \epsilon_i$$

where ϵ_i is the error term.

- Polynomial regression extends the linear model by adding extra predictors, obtained by raising each of the original predictors to a power.
- Least squares returns variance estimates for each of the fitted coefficients β_j^2 as well as the covariances between pairs of coefficient estimates.

```
library (ISLR2)
attach (Wage)
fit <- lm(wage ~ poly (age , 4), data = Wage)
coef ( summary (fit))

##                   Estimate Std. Error    t value     Pr(>|t|)
## (Intercept)    111.70361  0.7287409 153.283015 0.000000e+00
## poly(age, 4)1  447.06785 39.9147851  11.200558 1.484604e-28
## poly(age, 4)2 -478.31581 39.9147851 -11.983424 2.355831e-32
## poly(age, 4)3  125.52169 39.9147851   3.144742 1.678622e-03
## poly(age, 4)4  -77.91118 39.9147851  -1.951938 5.103865e-02

fit2 <- lm(wage ~ poly (age , 4, raw = T), data = Wage)
coef ( summary (fit2))

##                              Estimate   Std. Error   t value     Pr(>|t|)
## (Intercept)             -1.841542e+02 6.004038e+01 -3.067172 0.0021802539
## poly(age, 4, raw = T)1   2.124552e+01 5.886748e+00  3.609042 0.0003123618
## poly(age, 4, raw = T)2 -5.638593e-01 2.061083e-01 -2.735743 0.0062606446
## poly(age, 4, raw = T)3   6.810688e-03 3.065931e-03  2.221409 0.0263977518
## poly(age, 4, raw = T)4 -3.203830e-05 1.641359e-05 -1.951938 0.0510386498

fit2a <- lm(wage ~ age + I(age^2) + I(age^3) + I(age^4), data = Wage)
coef (fit2a)
```

```
##   (Intercept)            age       I(age^2)       I(age^3)       I(age^4)
## -1.841542e+02  2.124552e+01 -5.638593e-01  6.810688e-03 -3.203830e-05
```

```
fit2b <- lm(wage ~ cbind (age , age^2, age^3, age^4), data = Wage)
agelims <- range (age)
age.grid <- seq (from = agelims[1], to = agelims [2])
preds <- predict (fit , newdata = list (age = age.grid), se = TRUE)
se.bands <- cbind (preds$fit + 2 * preds$se.fit , preds$fit - 2 * preds$se.fi
t)
library(ggplot2)
ggplot()+
  geom_point(aes(Wage$age , Wage$wage),col="pink")+
  geom_smooth(aes(Wage$age , Wage$wage),col="red")
```

```
## `geom_smooth()` using method = 'gam' and formula = 'y ~ s(x, bs = "cs")'
```


```
  labs(title = "Degree -4 Polynomial" , x= "age" , y="Wage")
```

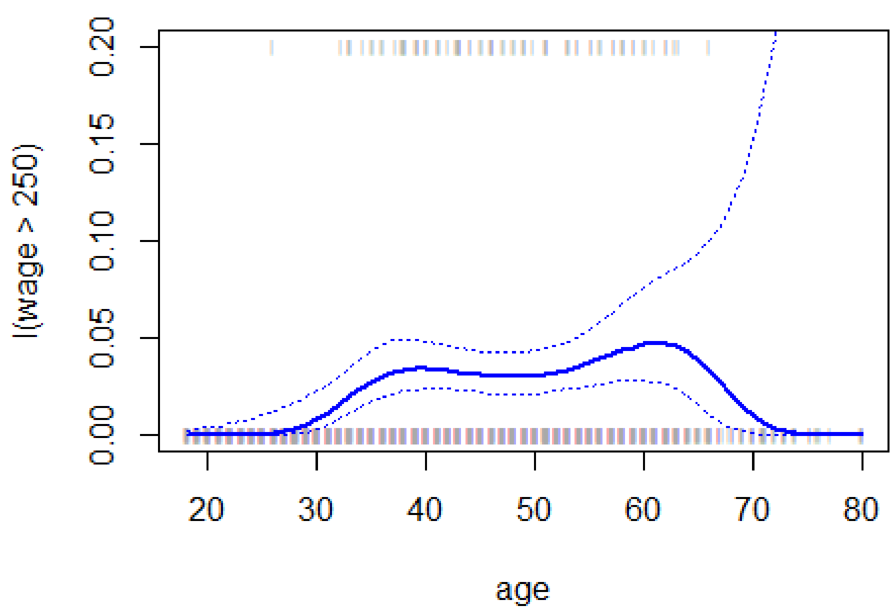
```
## $x
## [1] "age"
##
## $y
## [1] "Wage"
##
## $title
## [1] "Degree -4 Polynomial"
##
```

```
## attr(,"class")
## [1] "labels"
```

```
preds2 <- predict (fit2 , newdata = list (age = age.grid), se = TRUE)
max (abs (preds$fit - preds2$fit))
```

```
## [1] 7.81597e-11
```

```
fit <- glm (I(wage > 250) ~ poly (age , 4), data = Wage , family = binomial)
preds <- predict (fit , newdata = list (age = age.grid), se = T)
pfit <- exp (preds$fit) / (1 + exp (preds$fit))
se.bands.logit <- cbind (preds$fit + 2 * preds$se.fit ,
preds$fit - 2 * preds$se.fit)
se.bands <- exp (se.bands.logit) / (1 + exp (se.bands.logit))
preds <- predict (fit , newdata = list (age = age.grid), type = "response", s
e = T)
plot (age , I(wage > 250), xlim = agelims , type = "n", ylim = c(0, .2))
points ( jitter (age), I((wage > 250) / 5), cex = .5, pch = "|", col = "darkg
rey")
lines (age.grid, pfit , lwd = 2, col = "blue")
matlines (age.grid , se.bands , lwd = 1, col = "blue", lty = 3)
```




```
table ( cut (age , 4))
```

```
##
## (17.9,33.5]   (33.5,49]   (49,64.5] (64.5,80.1]
##         750        1399         779          72
```

```
fit <- lm(wage ~ cut (age , 4), data = Wage)
coef(summary (fit))

##                          Estimate Std. Error   t value     Pr(>|t|)
## (Intercept)             94.158392   1.476069 63.789970 0.000000e+00
## cut(age, 4)(33.5,49]    24.053491   1.829431 13.148074 1.982315e-38
## cut(age, 4)(49,64.5]    23.664559   2.067958 11.443444 1.040750e-29
## cut(age, 4)(64.5,80.1]   7.640592   4.987424  1.531972 1.256350e-01
```

Step Functions:

Step functions cut the range of a variable into K distinct regions in order to produce a qualitative variable. We create cutpoints $c_1, c_2, \ldots, c_K$ in the range of X, and then construct $K + 1$ new variables.

$$\begin{aligned}
C_0(X) &= I(X < c_1) \\
C_1(X) &= I(c_1 \leq X < c_2) \\
C_2(X) &= I(c_2 \leq X < c_3) \\
&\vdots \\
C_{K-1}(X) &= I(c_{K-1} \leq X < c_K) \\
C_K(X) &= I(c_K \leq X)
\end{aligned}$$

We then use least squares to fit the linear model

$$y_i = \beta_0 + \beta_1 C_1(x_i) + \beta_2 C_2(x_i) + \beta_3 C_3(x_i) + \cdots + \beta_K C_K(x_i) + \epsilon_i$$

Regression Splines:

- Regression splines are more flexible than polynomials and step functions, and in fact are an extension of the two. They involve dividing the range of X into K distinct regions. Within each region, a polynomial function is fit to the data. However, these polynomials are constrained so that they join smoothly at the region boundaries, or knots.

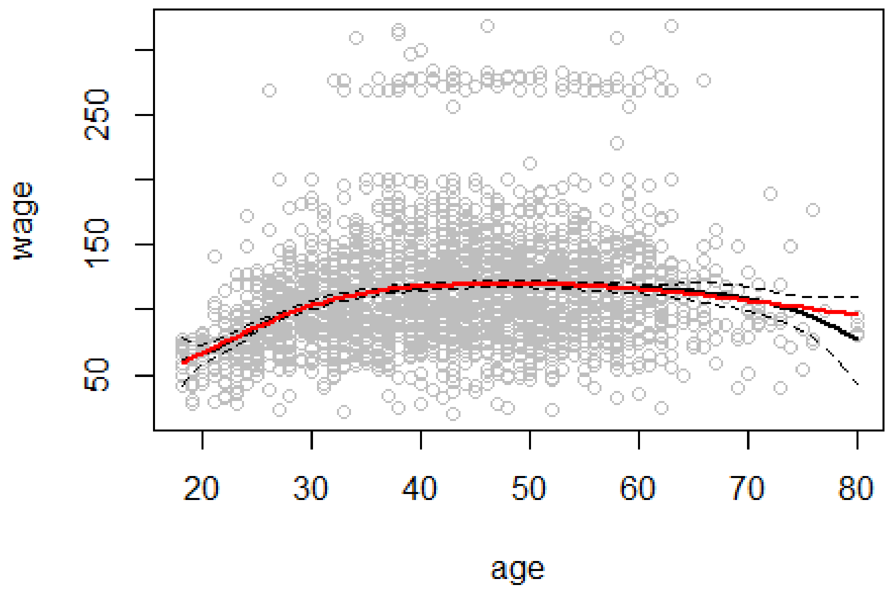
A piecewise cubic polynomial with a single knot at a point c takes the form

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

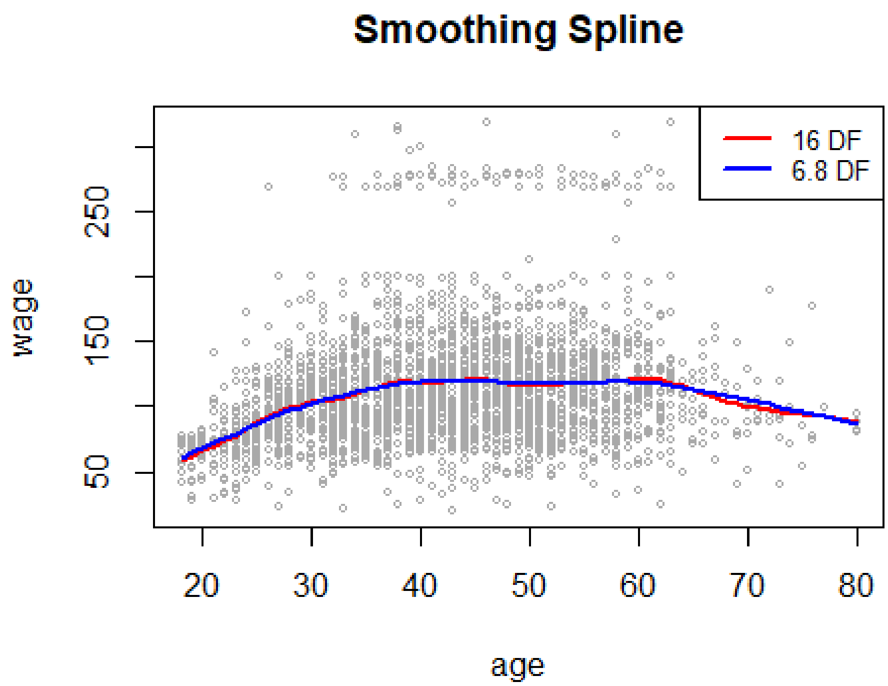
- To choose number of knots an approach is to use cross-validation. With this method, we remove a portion of the data (say 10 %), fit a spline with a certain number of knots to the remaining data and then use the spline to make predictions for the held-out portion.

```
library (splines)
fit <- lm(wage ~ bs(age , knots = c(25, 40, 60)), data = Wage)
pred <- predict (fit , newdata = list (age = age.grid), se = T)
plot (age , wage , col = "gray")
lines (age.grid, pred$fit , lwd = 2)
lines (age.grid , pred$fit + 2 * pred$se, lty = "dashed")
lines (age.grid , pred$fit - 2 * pred$se, lty = "dashed")
fit2 <- lm(wage ~ ns(age , df = 4), data = Wage)
```

```
pred2 <- predict (fit2 , newdata = list (age = age.grid), se = T)
lines (age.grid , pred2$fit , col = "red", lwd = 2)
```




```
plot (age , wage , xlim = agelims , cex = .5, col = "darkgrey")
title ("Smoothing Spline")
fit <- smooth.spline (age , wage , df = 16)
fit2 <- smooth.spline (age , wage , cv = TRUE)

## Warning in smooth.spline(age, wage, cv = TRUE): cross-validation with non-
unique
## 'x' values seems doubtful

fit2$df

## [1] 6.794596

lines (fit , col = "red", lwd = 2)
lines (fit2 , col = "blue", lwd = 2)
legend ("topright", legend = c("16 DF", "6.8 DF") , col = c("red", "blue"), l
ty = 1, lwd = 2, cex = .8)
```



Smoothing Splines:

Smoothing splines result from minimizing a residual sum of squares criterion subject to a smoothness penalty. We really want is a function g that makes RSS small but that is also smooth. A natural approach is to find the function g that minimizes

$$\sum_{i=1}^{n}\left(y_i - g(x_i)\right)^2 + \lambda \int g''(t)^2 dt$$

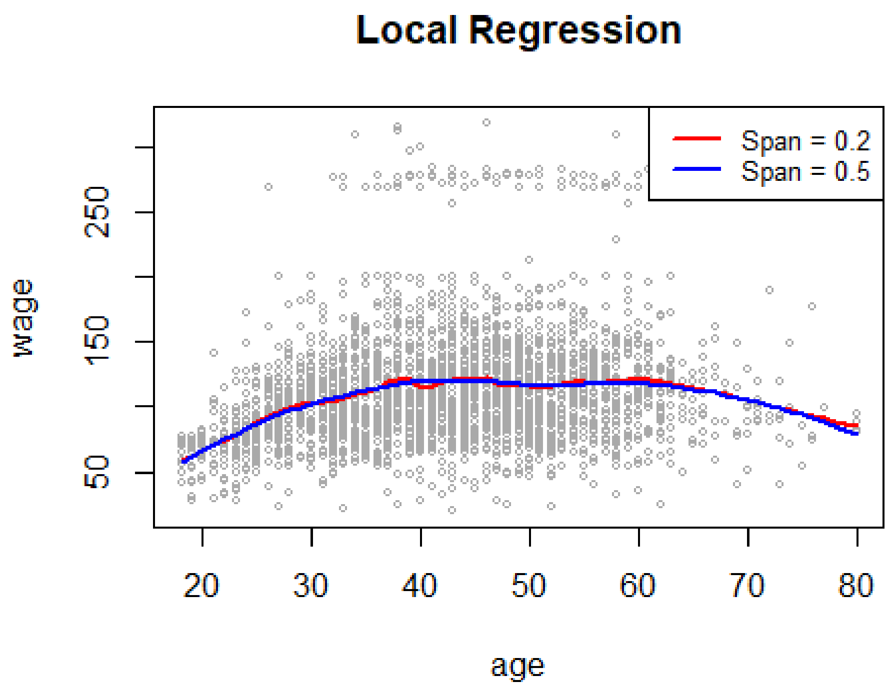
where λ is a nonnegative tuning parameter and chosen using LOOCV. Ihe equation takes the "Loss+Penalty" formulation.

Local Regression:

- In Local regression the regions are allowed to overlap, and indeed they do so in a very smooth way.

1. Gather the fraction $s = k/n$ of training points whose xi are closest to x_0.
2. Assign a weight $K_{i0} = K(x_i, x_0)$ to each point in this neighborhood, so that the point furthest from x_0 has weight zero, and the closest has the highest weight. All but these k nearest neighbors get weight zero.
3. Fit a weighted least squares regression of the y_i on the x_i using the aforementioned weights, by finding $\widehat{\beta_0}$ and $\widehat{\beta_1}$ that minimize $\sum_{i=1}^{n} K_{i0}\,(y_i - \beta_0 - \beta_1 x_i)^2$.
4. The fitted value at x_0 is given by $\hat{f}(x_0) = \hat{\beta}_0 + \hat{\beta}_1 x_0$

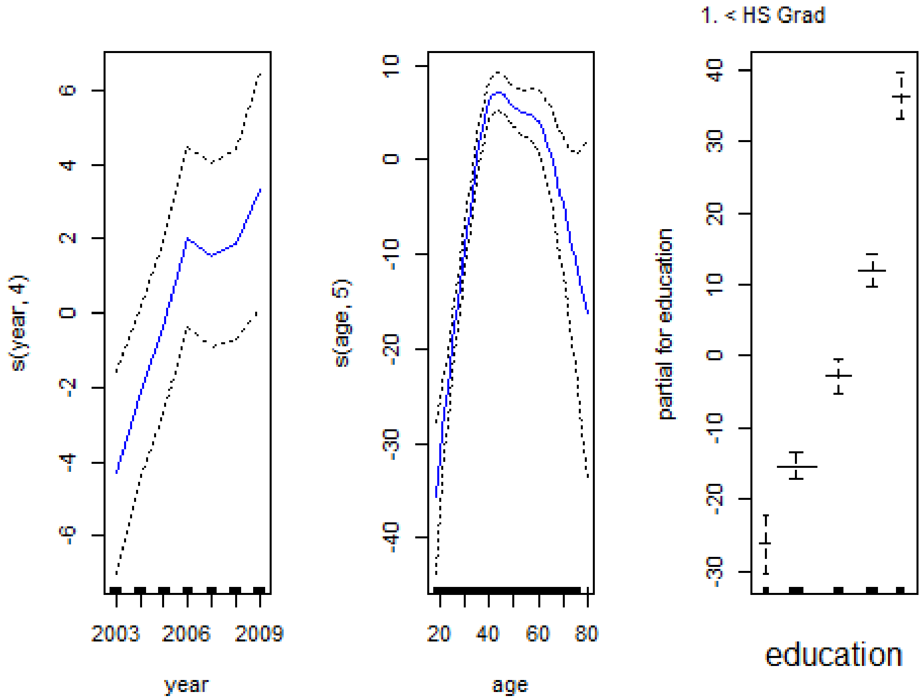
```r
plot (age , wage , xlim = agelims , cex = .5, col = "darkgrey")
title ("Local Regression")
fit <- loess (wage ~ age , span = .2, data = Wage)
fit2 <- loess (wage ~ age , span = .5, data = Wage)
lines (age.grid, predict (fit , data.frame (age = age.grid)),
col = "red", lwd = 2)
lines (age.grid, predict (fit2 , data.frame (age = age.grid)),
col = "blue", lwd = 2)
legend ("topright", legend = c("Span = 0.2", "Span = 0.5") ,
col = c("red", "blue"), lty = 1, lwd = 2, cex = .8)
```



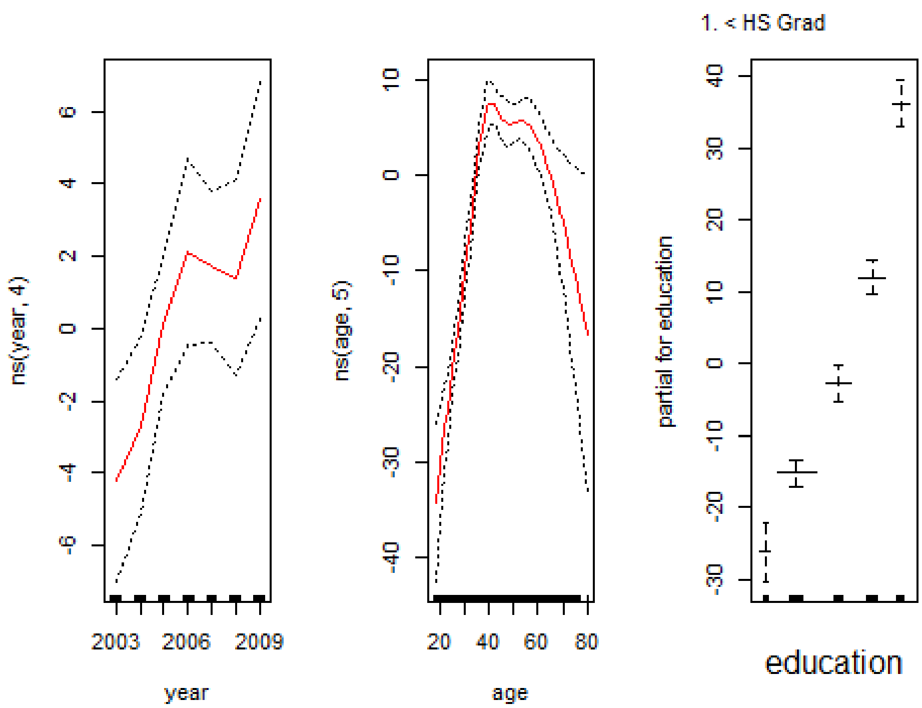
Generalized Additive Models:

Generalized additive models (GAMs) provide a general framework for extending a standard linear model by allowing non-linear functions of each of the variables, while maintaining additivity.

```r
gam1 <- lm(wage ~ ns(year , 4) + ns(age , 5) + education , data = Wage)
library (gam)

## Loading required package: foreach

## Loaded gam 1.22

gam.m3 <- gam (wage ~ s(year , 4) + s(age , 5) + education , data = Wage)
par (mfrow = c(1, 3))
plot (gam.m3, se = TRUE , col = "blue")
```




```
plot.Gam (gam1 , se = TRUE , col = "red")
```



```r
gam.m1 <- gam (wage ~ s(age , 5) + education , data = Wage)
gam.m2 <- gam (wage ~ year + s(age , 5) + education , data = Wage)
anova (gam.m1 , gam.m2 , gam.m3 , test = "F")
```

```
## Analysis of Deviance Table
##
## Model 1: wage ~ s(age, 5) + education
## Model 2: wage ~ year + s(age, 5) + education
## Model 3: wage ~ s(year, 4) + s(age, 5) + education
##   Resid. Df Resid. Dev Df Deviance       F    Pr(>F)
## 1      2990    3711731
## 2      2989    3693842  1  17889.2 14.4771 0.0001447 ***
## 3      2986    3689770  3   4071.1  1.0982 0.3485661
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```