STAT 621 Lecture Notes
Nonparametric Regression Continued: Splines

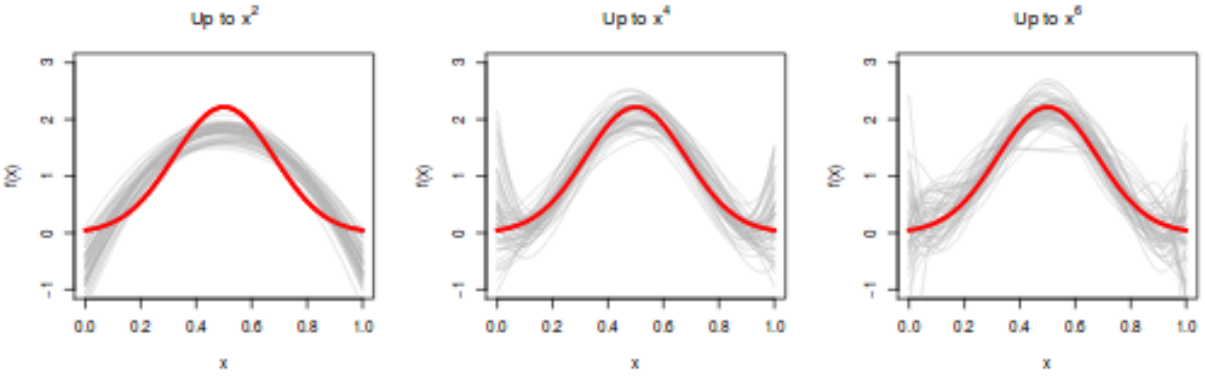
Next we consider a slightly different approach to estimating a regression function nonparametrically. Our discussion here will loosely follow that given in Chapter 5 of Hastie et al. *The Elements of Statistical Learning* or Chapter 7 of *Introduction to Statistical Learning*. This approach uses a *linear basis expansion* of the regression function. First recall our assumed regression model between Y and a single predictor X,

Typically the regression function $g(x)$ is some unknonwn, nonlinear function. The strategy here is to model $g(x)$ as a linear combination of basis functions. That is we assume

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

where the $h_m(X)$ are known functions. For example, if we let $h_m(X) = X^m$, what is our model?

As we've seen once or twice, polynomial models may be good at capturing some nonlinear features, but they they may exhibit undesirable behaviour, especially near data boundaries. The plot below (taken from Patrick Breheny's Univ. KY materials) illustrates this.



Piecewise Polynomials

One way around this is difficulty is to model $g(x)$ using piecewise polynomials. We break the x-axis separate ranges by defining a set of *knots*, $a_1, a_2, \ldots, a_K$. Separate polynomials are fit to the data in each section.

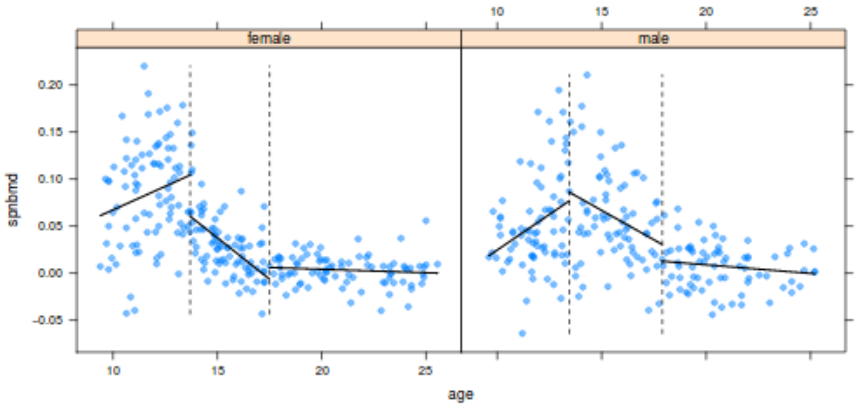
Example: Piecewise constant model. Consider the following basis functions.

$$h_1(x) = I(x < a_1) \qquad h_2(x) = I(a_1 \leq x < a_2) \qquad h_3(x) = I(a_2 \leq x)$$

What is our model for $g(x)$? What will the estimated parameters be? Draw a sketch.

Example: Write down a set of basis functions that will model $g(x)$ as a piecewise linear function.

Here's an example modeling spinal measurements of bone minearal density as a function of age for male and female adolescents (again from Patrick Breheny). Obviously this does not make an ideal model.



Spline Models: These models extend the idea of piecewise polynomials by imposing restrictions on the pieces. These restrictions include

(1) Constraints to make the estimated function meet at the knots. An example of such a constraint with our piecewise linear model:

(2) Constraints to make the 1st and/or 2nd, etc. derivatives of the estimated function continuous at the knots. What will this do? (Figure from Hastie et al. *ESL*)

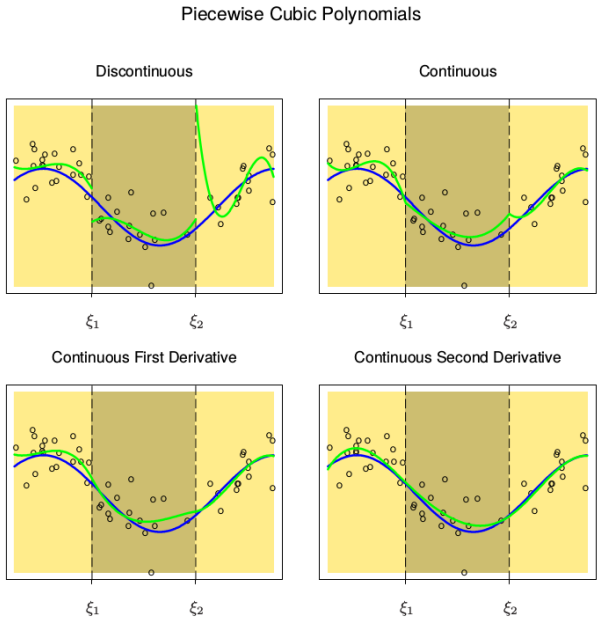


FIGURE 5.2. *A series of piecewise-cubic polynomials, with increasing orders of continuity.*

Commonly Used Spline Models. Some spline models and basis functions are common choices. These include

- Natural Cubic Splines: These models use piecewise cubic functions that are constrained to be continuous at the knots, and to have continuous 1st and 2nd derivatives. In addition, to alleviate poor boundary behavior, the function is forced to be linear outside the knot boundaries.
- B-Spline Basis Models: These models use the B-spline basis functions. These are more complicated basis functions, but they have superior properties in many cases. They tend to be numerically more stable.
- Thin Plate Splines: These use basis functions that are defined in two (or maybe more) dimensions. They are often used for estimating functions that are defined over spatial regions.

More Details

- How to select knot locations?

- How do we choose the number of knots?

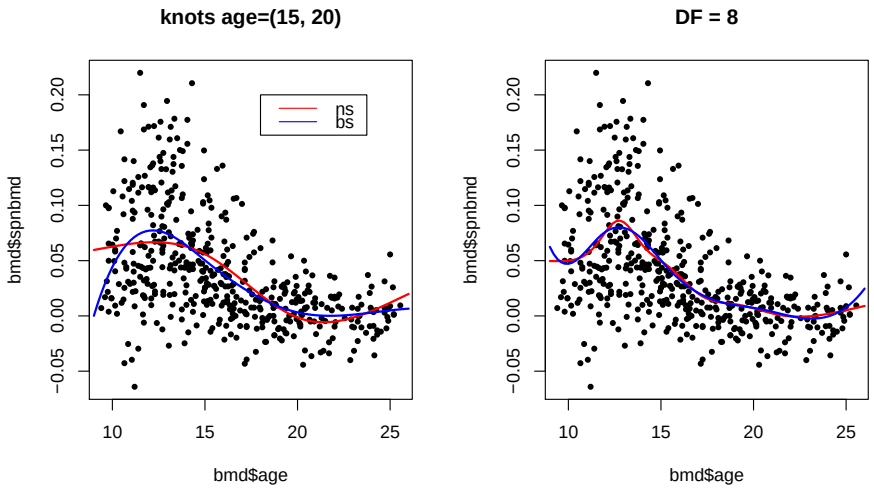
- **Smoothing Splines** to avoid some of these issues.

- Rules for Degrees of Freedom.

- Inferences with Splines.

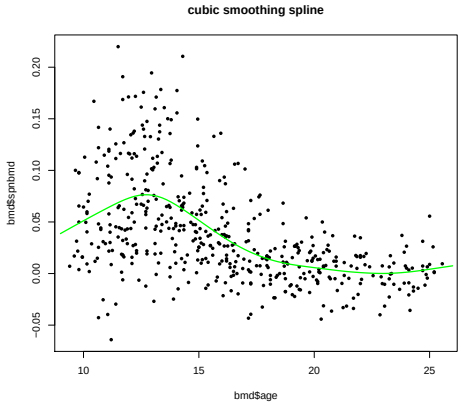
Example: This example uses the bone mineral density data. The R function `ns` is used to fit models using the natural cubic splines. The similar function `bs` is used to fit models using the B-spline basis. The function `smooth.spline` computes smoothing spline models.

```
 bmd=read.table(x,header=T)
 head(bmd)
  idnum   age gender      spnbmd
1     1 11.70   male 0.018080670
2     1 12.70   male 0.060109290
3     1 13.75   male 0.005857545
4     2 13.25   male 0.010263930
... ETC...

library(splines)
plot(bmd$age, bmd$spnbmd, cex=.75, pch=16)
age.pred=seq(9,26,.2)

# fit by specifying 2 knots
fit.ns=lm(spnbmd~ns(age,knots=c(15,20)),data=bmd)
lines(age.pred, predict(fit.ns, data.frame(age=age.pred)),col='red',lwd=2)
fit.bs=lm(spnbmd~bs(age,knots=c(15,20)),data=bmd)
lines(age.pred, predict(fit.bs, data.frame(age=age.pred)),col='blue',lwd=2)

title("knots age=(15, 20)")
legend(18,.2,c("ns","bs"),lty=c(1,1), col=c('red','blue'))


# fit by specifying df
fit.ns2=lm(spnbmd~ns(age,df=8),data=bmd)
plot(bmd$age, bmd$spnbmd, cex=.75, pch=16)
lines(age.pred, predict(fit.ns2, data.frame(age=age.pred)),col='red',lwd=2)
fit.bs2=lm(spnbmd~bs(age,df=8),data=bmd)
lines(age.pred, predict(fit.bs2, data.frame(age=age.pred)),col='blue',lwd=2)
title("DF = 8")
```



Here is the smoothing spline fit to the same dataset. The smoothing parameter is chosen with generalized cross validation (`cv=TRUE` does CV). The option `all.knots=T` places knots at each observed age value. Alternatively you can specify `nknots=`.

```
fit.ss=smooth.spline(bmd$age, bmd$spnbmd, cv=F, all.knots=T)
plot(bmd$age, bmd$spnbmd, cex=.75, pch=16)
pred.ss=predict(fit.ss, age.pred)
lines(pred.ss$x,pred.ss$y, lwd=2, col='green')
title("cubic smoothing spline")
```



Other Basis Approaches: