5.2 Polynomial regression and step functions

Polynomial regression and step functions

Polynomial regression and step functions

In regression problems the underlying function $f(X)$ is typically nonlinear and non-additive in X. However, representing $f(X)$ by a linear model is a convenient and sometimes necessary approximation:

- **Convenient**: linear model is easy to interpret, it is a first-order Taylor approximation to $f(X)$;
- **Necessary**: with N small and/or p large, linear model might be all we can fit to the data without overfitting.

Polynomial regression extends the linear model by adding extra predictors, obtained by raising each of the original predictors to a power.

The core idea in polynomial regression (and other nonlinear regression methods discussed in this week's course material) is to augment the vector of input X with additional variables, which are transformations of X, and use linear models in this new space of derived input features.

For simple linear regression, the design matrix is $\boldsymbol{X} = \begin{bmatrix} 1 & x_1 \\ \vdots & \vdots \\ 1 & x_n \end{bmatrix}$; we say that $1, x$ is a **basis**.

For a quadratic model, $y_i = \beta_0 + \beta_1 x_i + \beta_2 x_i^2 + \varepsilon_i$ the design matrix is

$X = \begin{bmatrix} 1 & x_1 & x_1^2 \\ \vdots & \vdots & \vdots \\ 1 & x_n & x_n^2 \end{bmatrix}$ and the basis $1, x, x^2$.

Example: Polynomial regression

In this example we analyze the `Wage` data from the `ISLR` library.

```
library(ISLR)
data("Wage")
attach(Wage)

fit=lm(wage~poly(age,4,raw=T), data=Wage)
coef(summary(fit))
```

```
                            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
```

There are several other equivalent ways of fitting this model:

```
library(ISLR)
data("Wage")
attach(Wage)

fit2=lm(wage~age+I(age^2)+I(age^3)+I(age^4),data=Wage)
coef(summary(fit2))
```

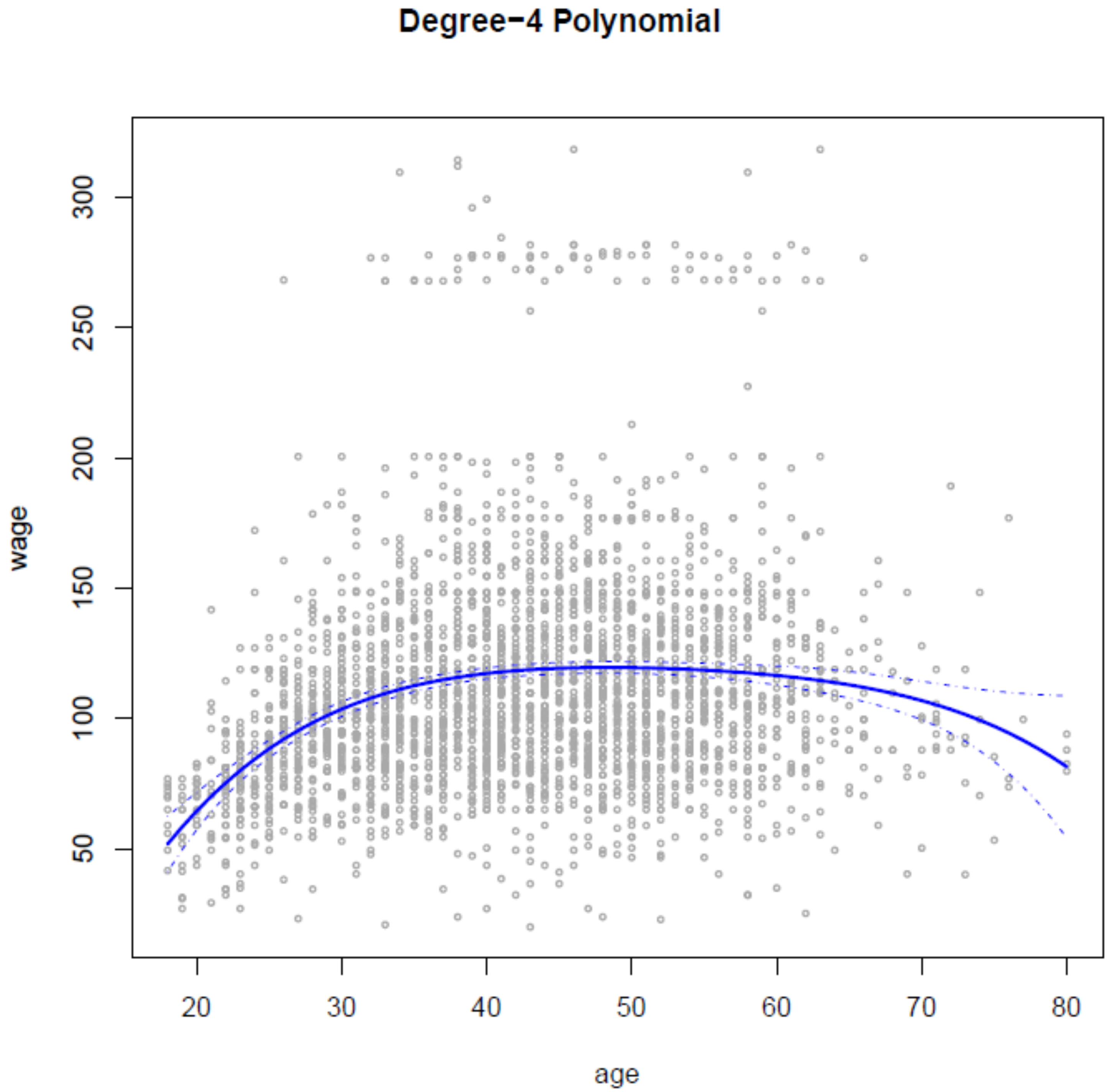
```
                 Estimate   Std. Error   t value     Pr(>|t|)
(Intercept) -1.841542e+02 6.004038e+01 -3.067172 0.0021802539
age          2.124552e+01 5.886748e+00  3.609042 0.0003123618
I(age^2)    -5.638593e-01 2.061083e-01 -2.735743 0.0062606446
I(age^3)     6.810688e-03 3.065931e-03  2.221409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

or

```
library(ISLR)
data("Wage")
attach(Wage)

fit3=lm(wage~cbind(age,age^2,age^3,age^4),data=Wage)
```

Let us now fit the degree-4 polynomial to `wage` as a function of `age` in the `Wage` dataset.

```
library(ISLR)
data("Wage")
attach(Wage)

fit=lm(wage~poly(age,4,raw=T), data=Wage)
attach(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.fit)
par(mfrow=c(1,1),mar=c(4.5,4.5,1,1),oma=c(0,0,4,0))
plot(age,wage,xlim=agelims, cex=.5,col="darkgrey")
title("Degree-4 Polynomial", outer=T)
lines(age.grid, preds$fit, lwd=2, col="blue")
matlines(age.grid, se.bands, lwd=1,col="blue",lty=4)
```



Below we also illustrate one way of choosing the degree of the polynomial to use. We now fit models ranging from linear to a degree-5 polynomial and use hypothesis tests to determine the simplest model which is sufficient to explain the relationship between `wage` and `age`.

```
library(ISLR)
data("Wage")
attach(Wage)

fit.1=lm(wage~age,data=Wage)
fit.2=lm(wage~poly(age,2),data=Wage)
fit.3=lm(wage~poly(age,3),data=Wage)
fit.4=lm(wage~poly(age,4),data=Wage)
fit.5=lm(wage~poly(age,5),data=Wage)
anova(fit.1,fit.2,fit.3,fit.4,fit.5)
```

```
Analysis of Variance Table

Model 1: wage ~ age
Model 2: wage ~ poly(age, 2)
Model 3: wage ~ poly(age, 3)
Model 4: wage ~ poly(age, 4)
Model 5: wage ~ poly(age, 5)
  Res.Df     RSS Df Sum of Sq        F    Pr(>F)
1   2998 5022216
2   2997 4793430  1    228786 143.5931 < 2.2e-16 ***
3   2996 4777674  1     15756   9.8888  0.001679 **
4   2995 4771604  1      6070   3.8098  0.051046 .
5   2994 4770322  1      1283   0.8050  0.369682
---
Signif . codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The p-value corresponding to comparing linear $\texttt{Model1}$ to the quadratic $\texttt{Model2}$ is small indicating that a linear fit is not sufficient. The p-value corresponding to comparing $\texttt{Model2}$ to $\texttt{Model3}$ is also low indicating that $\texttt{Model2}$ is not sufficient. The p-value corresponding to comparing $\texttt{Model3}$ and $\texttt{Model4}$ is greater than 0.05 indicating that there is no sufficient improvement in choosing degree-4 polynomial over the cubic polynomial. Hence the cubic polynomial appears to provide a reasonable fit to the data.

Note that as an alternative to using ANOVA, we could choose the polynomial degree using cross-validation.

Example: polynomial logistic regression

We consider a procedure for predicting whether an individual earns more than $250,000 per year.

```
library(ISLR)
data("Wage")
attach(Wage)

fit=glm(I(wage>250)~poly(age,4),data=Wage,family=binomial)

agelims=range(age)
age.grid=seq(from=agelims[1], to=agelims[2])

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))
```

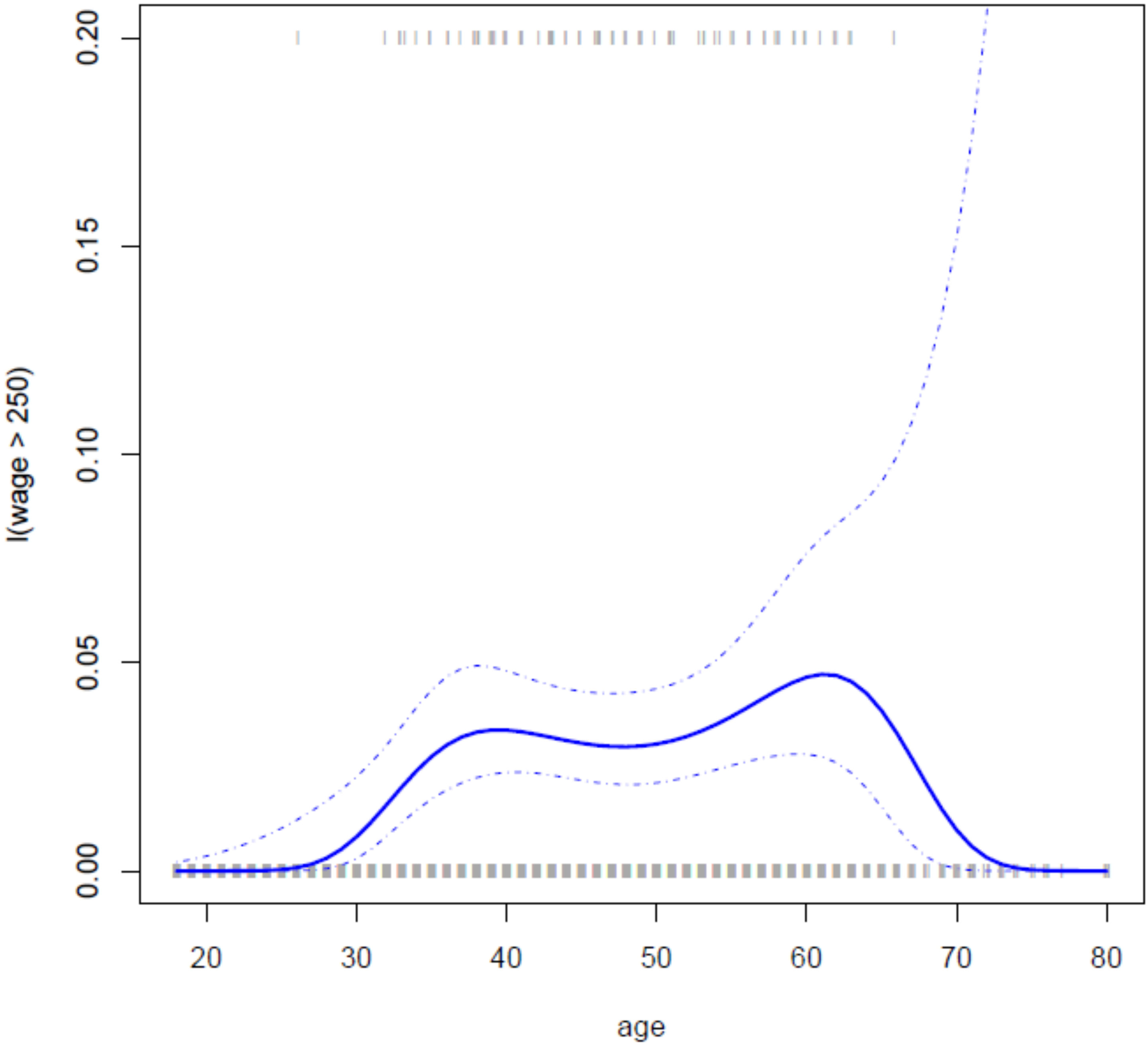
In order to directly compute the probabilities we select $\texttt{type} = \texttt{"response"}$ in $\texttt{predict}()$ as follows:

```
library(ISLR)
data("Wage")
```

```
attach(Wage)

fit=glm(I(wage>250)~poly(age,4),data=Wage,family=binomial)
agelims=range(age)
age.grid=seq(from=agelims[1], to=agelims[2])
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",se=T)
```

For visualization we execute the following code:

```
library(ISLR)
data("Wage")
attach(Wage)

fit=glm(I(wage>250)~poly(age,4),data=Wage,family=binomial)
agelims=range(age)
age.grid=seq(from=agelims[1], to=agelims[2])
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",se=T)

attach(Wage)

## The following objects are masked from Wage (pos = 4):
##
## age, education, health, health ins, jobclass, logwage, maritl,
## race, region, sex, wage, X, year

plot(age, I(wage>250),xlim=agelims, type="n", ylim=c(0,0.2))
points(jitter(age), I((wage>250)/5), cex=.5,pch="|",col="darkgrey")
lines(age.grid,pfit, lwd=2, col="blue")
matlines(age.grid, se.bands, lwd=1, col="blue", lty=4)
```



Note the wide confidence intervals on the right-hand side. Although the sample size for this data is large ($3,000$), there are only 79 high earners, which results in high variance in the estimated coefficients.

Linear basis expansion

Denote by $h_m(X) : \boldsymbol{R}^p \rightarrow \boldsymbol{R}$ the mth transformation of X, $m = 1, \ldots, M$. We then model

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

which is a **linear basis expansion** in X.

Note: once basis functions h_m are established, the models are **linear in these new variables** and the fitting proceeds as before.

Examples of basis functions:

- $h_m(X) = X_m$, $m = 1, \ldots, p$ - original linear model;
- $h_m(X) = X_j^2$ - polynomial terms;
- $h_m(X) = \log(X_j), \sqrt{X_j}, \ldots$ - other nonlinear transformations of single input;
- $h_m(X) = I(L_m < X_k \leq U_m)$, and indicator for a region of X_k - model with piecewise constant contribution for X_k.

The last example of basis function brakes the range of X into **bins**, and fits a different constant in each bin.

Example: step functions

This example illustrates how to fit a step function in **R** using **cut**():

```
library(ISLR)
data("Wage")
attach(Wage)

table(cut(age,4))

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

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

```
                        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
```

Note that the **cut**() function returns an ordered categorical variable and **lm**() function creates a set of dummy variables for use in the regression.

Activity in R: Polynomial regression

In this activity, you will further analyse the `Wage` data set considered in this section. Perform polynomial regression to predict `wage` using `age`. Use cross-validation to select the optimal degree d for the polynomial. What degree was chosen, and how does this compare to the results of hypothesis testing using ANOVA? Make a plot of the resulting polynomial fit to the data.

Additional Activity

Question 1 *Submitted Feb 18th 2024 at 8:54:59 pm*

Which of the following are true about polynomial regression:

- [x] polynomial regression extends the linear model by adding extra predictors, obtained by raising of the original predictors to a power;
- [x] polynomial regression can be fitted by the \textit{lm()} funciton in R;
- [x] the degree of the best fitting polynomial can be determined by cross-validation.

Question 2 *Submitted Feb 18th 2024 at 8:55:12 pm*

For a quadratic model, $y_i = \beta_0 + \beta_1 x_i + \beta_2 x_i^2 + \varepsilon_i$, the basis is

- ( ) $1, x^2;$
- ( ) $1, x^2, x^2;$
- (•) $1, x, x^2.$

Question 3 *Submitted Feb 18th 2024 at 8:55:18 pm*

A step function can be fitted to data in R by:

- ( ) $\texttt{lm}(\texttt{y\~{}I}(\texttt{x}))$
- (•) $\texttt{lm}(\texttt{y\~{}cut}(\texttt{x}, 4))$
- ( ) $\texttt{lm}(\texttt{y}, \texttt{poly}(\texttt{x}, 4, \texttt{raw} = \texttt{T}))$