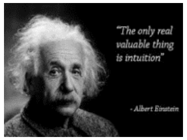
Spline (MARS) Models

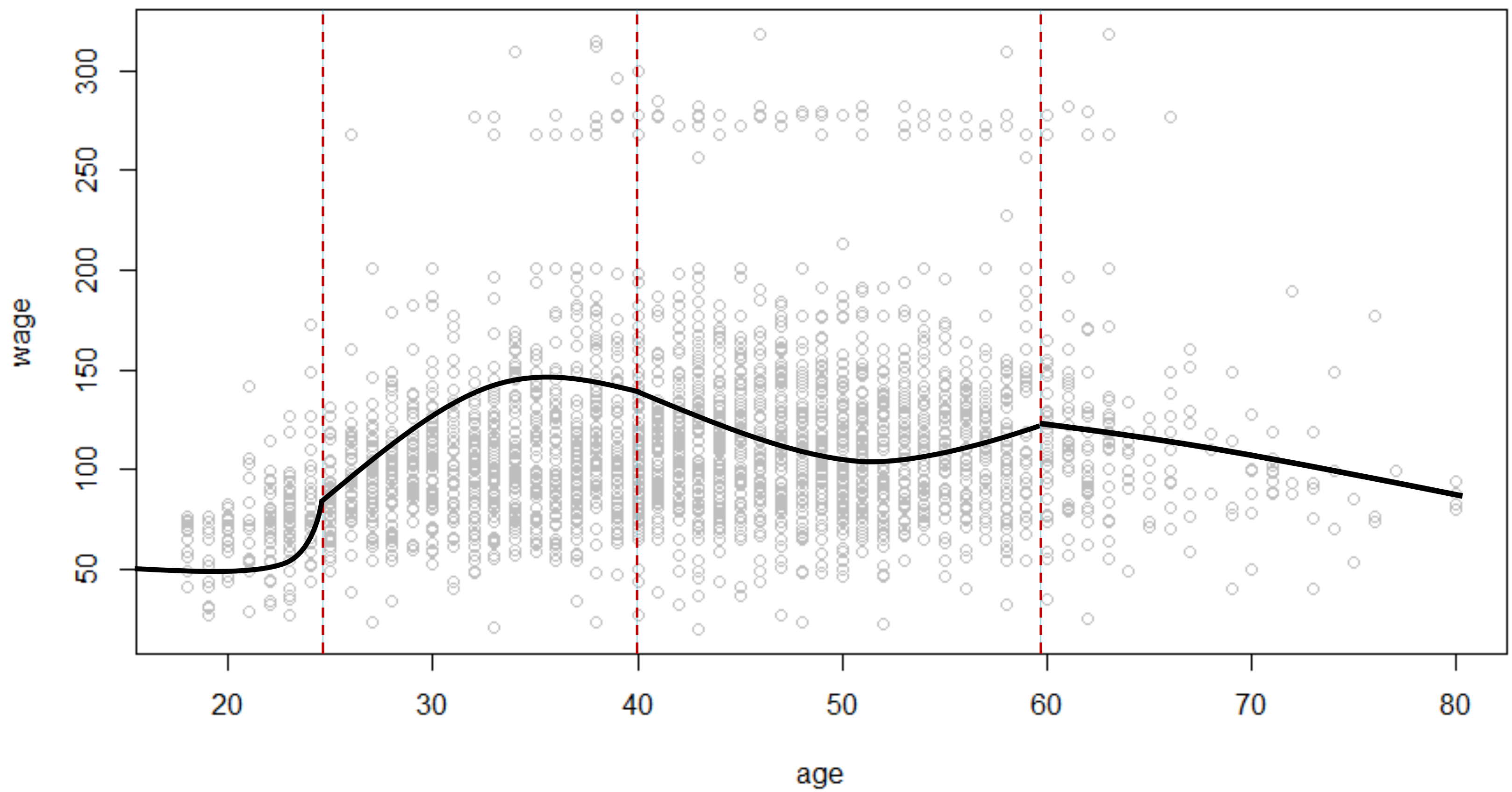
(Multivariate Adaptive Regression Spline)

KOGOD SCHOOL *of* BUSINESS
AMERICAN UNIVERSITY • WASHINGTON, DC



Spline Regressions: Intuition

A spline regression is very similar to a piecewise regression, except that an **additional constraint** is placed on the model so that **knots** have only **1 predicted value**, (i.e., **no shifts** at the knots) thus having a **continuous** function throughout the entire range of the predictor

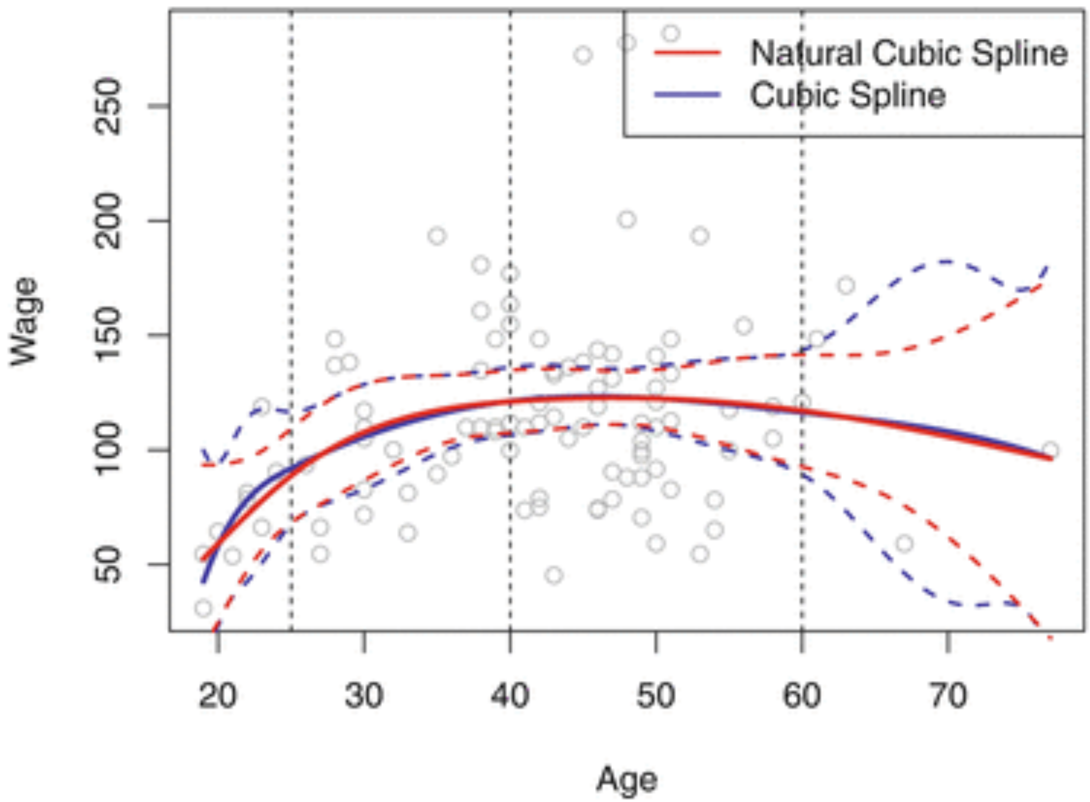


How Spline Regressions Work

- The full mathematics behind splines is explained in the textbook, but it boils down to a **simple concept**:
  - For a given variable, start at the **origin** with a plain **polynomial**
  - Then add a **"truncated" power function** for values beyond the **1st knot**:
    - ✓ 0 before the knot, i.e. for $x < c_1$ where c_1 is the 1st knot
    - ✓ A **polynomial** after the knot, i.e., $x >= c_1$ but using $x - c_1$ instead of x which will cause the two polynomials to connect at the dot
    - ✓ In the prior example, the truncated function for the first knot at $age = 25$ in a cubic spline would be:
      - ❖ 0 for $age < 25$
      - ❖ $(age-25)^3$ for $age >= 25$
  - Repeat this procedure at every knot
- This ensures that the various polynomial **segments connect** at the **knot**, thus yielding a **continuous** curve

Natural Splines

- Since we are using various polynomials to fit the splines, the splines will suffer from some of the typical **dimensionality** issues of polynomials, e.g., **high variance**, over-fitting
- Keeping the **dimensionality** of the segment splines **low** (i.e., cubic at the most) helps avoid some of these problems.
- However, splines are notorious for having **high variance** at the **outer ranges** of the data, where x is very **high** or very **low**.
- A **natural spline** helps correct for this problem to some extent by forcing the very **first** and **last segments** to a **linear** fit.



Tuning Spline Regressions

- As with other models, spline regressions need to be **tuned**
- The two most important tuning **parameters** are:

1. The number of **segments** or **knots** to use
    - More segments/knots add **complexity** to the model and make it harder to interpret
    - More **knots** will yield a tighter **training fit**, but will not necessarily improve the **test MSE**
    - Each **knot** uses up **1 degree of freedom** in the model, so be mindful before adding more knots.
    - **Knots** can be **evenly spaced** or specifically **selected** based on business knowledge or observations of plots
2. The **polynomial degree** or function to use in each segment
    - **Low degree** polynomials are preferred and more interpretable – i.e., no more than cubic

- As with most other methods the best models and tuning parameters should be evaluated with **cross-validation**

R Tips

`bs(){splines}` → "Spline Basis" function in the `{splines}` package to fit spline models

`fit.linear.spline1=lm(y~bs(x,knots=c(25,40,60),degree=1), data=dataName)` → Fits a linear spline (`degree=1`) with arbitrary knots at `x=24, 40` and `60`

`fit.linear.spline2=lm(y~bs(x,df=4,degree=1), data=dataName)` → Fits a linear spline (`degree=1`) with 4 equally spaced segments (`df=4`)

`ns(){splines}` → "Natural Cubic Spline" function in the `{splines}` package to fit natural cubic splines; degree does not need to be specified

`fit.natural=lm(y~ns(x,df=4),data=Wage)` → The ns() function fits a natural cubic spline with 4 segments

Kogod School
of
Business