Stat 435
Arnav Palkhiwala

1. Summary:

This week, we focused on models that can be used when our data is not linear. We learned about polynomial regression and how it can be used to model data in cubic/quadratic situations, as well as its downsides. Then we went on to learn about splines and knots in modeling functions, where there is different behavior for different x-values. Lastly, we covered GAMs and used them for more than 1 predictor situations.

2. Concepts:

Polynomial Regression:

- A downside of this is that it does not do well with tails
- Tends to not extrapolate well since there is not as much data

Step Function:

- Breaking down function by x and fitting different lines for each region
- Cut variable into distinct regions with indicator functions that are 1 if x is in a certain range or else 0
- Then run OLS on each cut

Basis Function Approach:

- Fit linear model WRT basis functions
- Can be split over different regions of x
- Splines allow for maximum amounts of continuity, knots are where polynomials change
- We don't want discontinuous functions and at knot x=c we want f1 at c to equal f2 at c
- Splines: make sure we have maximum smoothness
- If we are fitting degree k polynomials to each region we want the first to (k-1)'s derivative to be equal to 0
- Spline of degree k with knots t1 to tr is a piecewise polynomial with degree of k that is continuous
- Linear spline, piecewise linear and continuous at each point
- Cubic spline is commonly used, piecewise cubic polynomial with continuous derivative up to order
- Bias variance tradeoff with splines
    - For basic splines we can control degree k and number/location of knots
    - Increases degree k increases variance and decreases bias as the model is more complex

    - As you increase knots, variance increases and bias decreases because you are fitting each part to less data
- This is all chosen through CV
- Location of knots is determined: linearly spaced or quantiles (most used)
- Natural splines force polynomials rto have low degree to left and right at extreme knots
- (k-1)/2 on ends, forces continuity of derivatives as much as possible
- Smoothing splines:
    - Avoid having to choose number of knots and their locations
    - Solve for any tuning parameter lambda > 0
    - If lambda is very large solution is forced to be piecewise polynomial of degree m - 1
- Optimal f has closed form solution: degree 2m+1
- Generalized additive models are used for more than 1 predictor

3. Uncertainties:

In section we discussed how this works in R in terms of implementing these models, but I am curious about how we can do the same in Python. I have done all my homeworks in Python and typically there are functions to replicate R results in Python, but these topics seem more statistics-based so I am curious if it is more difficult in this situation. I am also curious about how interpretable some of the R outputs are for this because sometimes it is hard to understand how everything worked and how some of the knots were chosen, etc. I am also wondering how often splines are used because it seems like in real life situations you would not need many knots. So, at times would it be more effective to actually just create a piecewise function and break it down manually?