Non-linear Regression with Decision Trees and $k$NN
Decision Trees: A Review

To build the tree:

- Start with all observations at the root of the tree
- Score all remaining variables using the current set of observations
- Split the current set of observations based on the variable with the best score
- Repeat until all groups of observations are pure enough, or until the tree is sufficiently deep

To classify:

- Classify a new observation by walking the tree by its feature values
- Assign the new observation a class by majority vote, among all observations at the new observation’s leaf
Regression Trees

To build the tree:
- Start with all observations at the root of the tree
- Score all remaining variables using the current set of observations
- Split the current set of observations based on the variable with the best score
- Repeat until all groups of observations are pure enough, or until the tree is sufficiently deep

To classify:
- Classify a new observation by walking the tree by its feature values
- Assign the new observation a value by averaging the values of all observations at the new observation’s leaf
Regression Trees: Scoring

- To build decision trees, we scored variables based on an impurity measure. The more they distinguished among classes, the better!
- To build regression trees, we will instead score variables based on their variance. The lower the variance, the better!

<table>
<thead>
<tr>
<th>GOOD</th>
<th>BAD</th>
</tr>
</thead>
<tbody>
<tr>
<td>Score = var(1, 1.2) + var(2.2, 2.3) = 0.04</td>
<td>Score = var(1, 2.2) + var(1.2, 2.3) = 1.325</td>
</tr>
</tbody>
</table>
Regression Trees in R

Pretty much the same as decision trees!

`rpart` will detect whether you’re predicting real values or categorical ones.

```r
> attach(iris)
> rpart(Sepal.Length ~ Sepal.Width + Petal.Length + Petal.Width)
```
Regression with $k$NN

Same ideas as decision trees: Instead of majority vote, average!

- Average the values of all the neighbors equally
- Or, use a weighted average: assign greater weight to closer neighbors’ values
Unweighted

\[(3.2 + 1.3 + 0.5) / 3 = 1.6\]

Weighted

\[\frac{\frac{1}{1.2} \cdot 3.2 + \frac{1}{3} \cdot 1.3 + \frac{1}{4.3} \cdot 0.5}{\frac{1}{3} + \frac{1}{1.2} + \frac{1}{4.3}} = 2.3\]
Extras
Why are they called regression trees?

Key Idea: Instead of majority vote, regress at each leaf to find that node’s value!

How: Least squares: i.e., minimize the sum of the squared errors.

Q: What degree polynomial should we fit to the data? $d = 0, 1, 2$?
A: We prefer simpler models to more complex.

Next Q: What constant (i.e., $d = 0$) value minimizes the sum of the squared errors?
A: The average!
LO(W)ESS: $k$NN+

Locally (Weighted) Estimated Scatterplot Smoothing

In each neighborhood, run a regression! I.e., fit a polynomial.

Model selection: fix $k$, find the best $d$
- $d = 0$ yields a moving average (much like decision trees)
- $d = 1$ and $d = 2$ are also natural choices (i.e., linear or quadratic polynomial)

Model selection: fix $d$, find the best $k$
- $k$ must be at least $d + 1$: e.g., two points are needed to define a line