4.4: Nonparametric Regression

Dr. Bean - Stat 5100

1 Why nonparametric regression?

For most of this course, we have assumed models of the form:

$$Y_i = \beta_0 + \beta_1 X_{i,1} + \beta_2 X_{i,2} + \cdots + + \beta_{p-1} X_{i,p-1} + \epsilon.$$

Such models assume the following:

- Each explanatory variable shares a linear relationship with the response variable (perhaps aided by transformations).
  - In other words, after transformations, the rate of increase or decrease in Y is independent of the actual values of X.
- The effect of each explanatory variable can be isolated from the rest (assuming no interaction terms).
  - In other words, each explanatory variable is independent of all other explanatory variables.

(Groups) What are some consequences associated with inappropriately assuming a linear model?

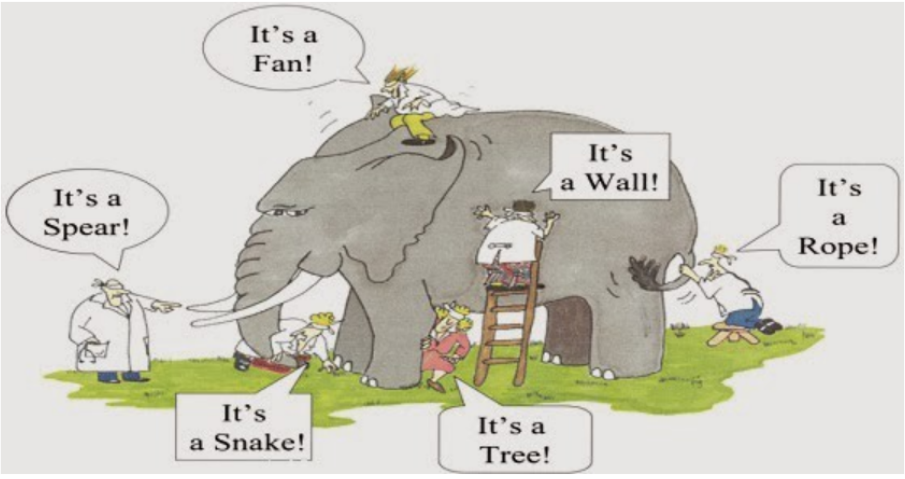

Figure 1: `https://medium.com/betterism/the-blind-men-and-the-elephant-596ec8a72a7d`

Nonparametric methods make far less (if any) assumptions about the form of the relationship between the explanatory and response variables.

The cost: Methods are often much more "data hungry" and harder to explain.

2 LOESS (local regression)

Close relative, lowess (local weighted regression scatter plot smoothing)

2.1 Assumptions

- Predictor variables are pre-selected
- The response function is "smooth." (i.e. small changes in any X_i, lead to relatively small changes in Y).
- Error terms are normal with constant variance.

2.2 Process

In order to make a prediction $\hat{Y}$ for a particular "X-profile" (i.e. combination of unique values for each explanatory variable)

1. (optional) standardize predictor variables X_i
2. For each observation i, calculate the distance to the current X-profile $X_{h,j}$

$$d_i = \sum_{j=1}^{p-1} (X_{i,j} - X_{h,j})^2$$

3. Let q = proportion of observations nearest to the current X-profile ($q \in (0, 1)$)
4. Let d_q = distance from X-profile to the furthest observation in the neighborhood as defined by q
5. For each observation i within that neighborhood, define weight

$$w_i = \begin{cases} \left(1 - \left(\frac{d_i}{d_q}\right)^3\right)^3 & d_i < d_q \\ 0 & otherwise \end{cases}$$

6. Using these weights, fit a weighted least squares (WLS) regression model based on polynomials of all predictors.
7. Use the WLS model to estimate $\hat{Y}$
    - Polynomial degree:
        - 0 - moving average
        - 1 - connected lines
        - 2 - smooth curves
        - (don't typically go higher than degree 2 as this can lead to unstable fits)

2.3 Implementation

LOESS requires the user to select the smoothing parameter q. (See Figure 2.)

- Larger q $\rightarrow$ smoother fit
- Smaller q $\rightarrow$ "choppy fit"

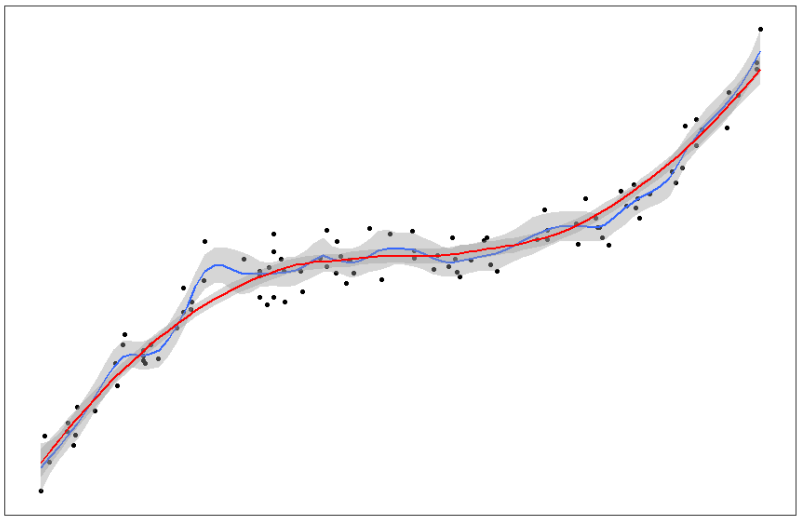

Figure 2: Example LOESS smoothing curves with only one X-variable and two levels of smoothness.

- Advantages
  - Flexible response surface - do not have to worry about whether or not the data share a linear relationship.
- Disadvantages:
  - Requires "dense" data to get good predictions.
    - Method extremely sensitive to outliers in "sparse" data regions.
  - No "model" to report - no inference.

In general, the less our model *assumes*, the more data we must *consume*.

3 Regression Trees

Simple, yet powerful way to handle high-ordered interactions between variables.

3.1 Process

- Separate the data into two **branches** by splitting the data in a way that minimizes the sum of squares error $\sum_i \left(Y_i - \hat{Y}_i\right)^2$ (or a similar metric).
  - Predictions $\hat{Y}_i$ in this case is the average of the values in each **terminal node or leaf** (i.e. the group of values that fall into each branch at the end of the tree).
- Keep splitting the subgroups over and over until all nodes are completely **pure** ($\sum_i \left(Y_i - \hat{Y}_i\right)^2 = $).
  - This may mean that each terminal node in the **fully grown** tree will be single observations.
- Because a model that perfectly predicts the training data is obviously overfit, we will **prune** the tree back to a set of cuts that balances accuracy with simplicity.
  - Typically picked using a **cost complexity parameter**:

$$CC(T) = R(T) + \alpha|T|$$

    - $CC(T)$ - cost complexity
    - $R(T)$ - error rate (such as average squared error)
    - α - user selected cost parameter (controls size of tree).
    - $|T|$ - number of nodes in the tree
  - Alternatively, complexity can be defined using restrictions on the tree such as:
    - Minimum number of observations in a terminal node.
    - Minimum percentage increase in the percent variance explained in order for a split to be conducted.

Example: predicting snow density using climate reanalysis data.
Variables

- maxv_SNWD - the depth of the snowpack (mm)
- TD - difference in the mean annual temperature between the coldest and the warmest month of the year (degrees Celsius)
- PPTWT - total winter (Dec to Feb) precipitation.

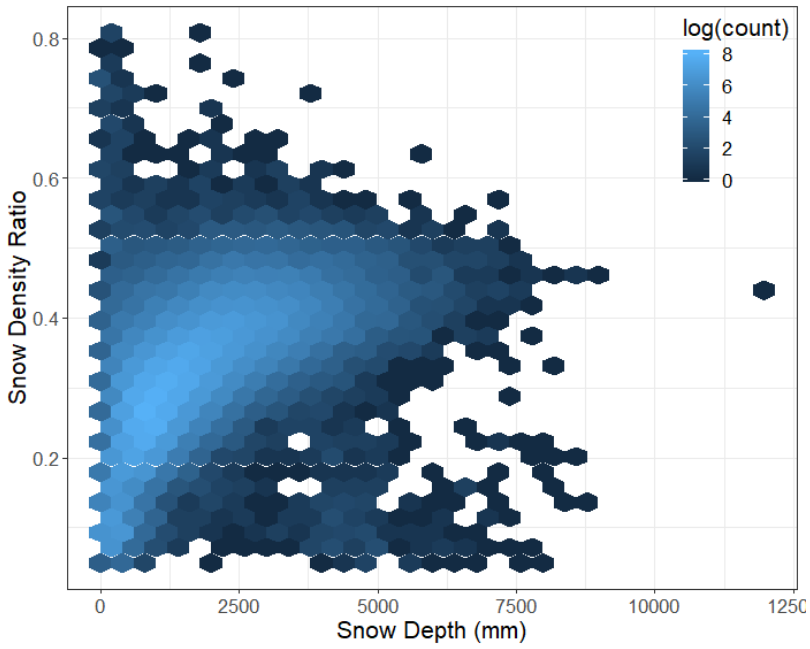

Figure 3: Plot of the snow density ratio in relation to its depth for locations across North America.

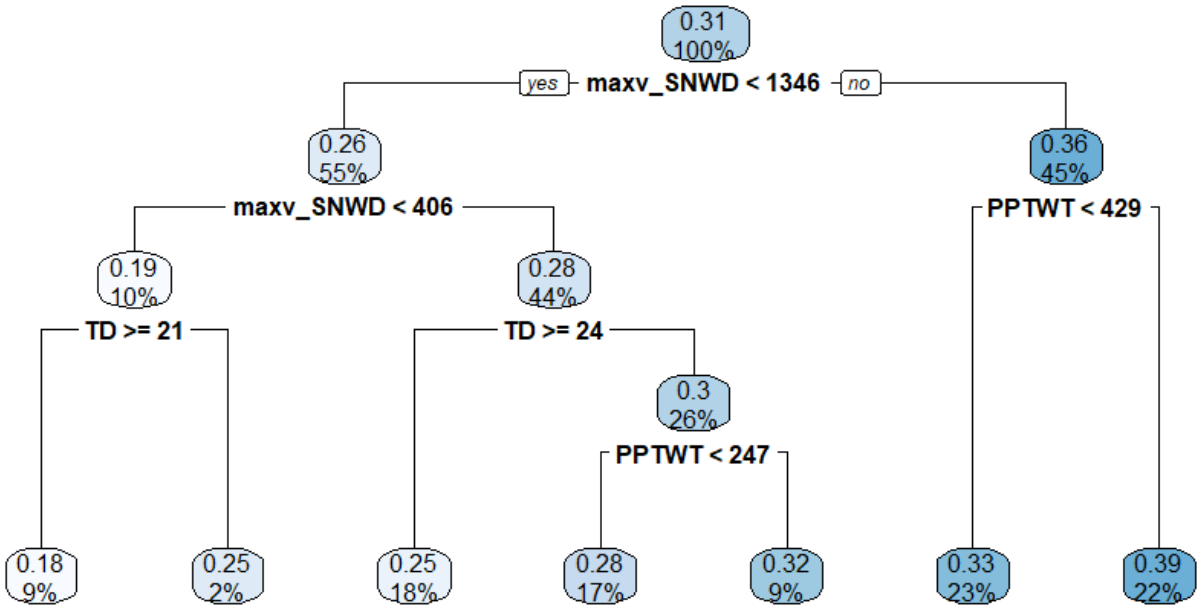

Figure 4: Sample tree (pruned) for predicting snow ratio using climate variables.

3.2 Variable Importance

There are several ways in which we can explore the importance of variables in a regression tree.

- **Count:** Variables that are used *more often* for splitting are more important.
- **Error Reduction:** The greater reduction in the SSE resulting from splitting on a variable, the more important a variable.

3.3 Extensions of Regression Trees

- Boosting - fit tree in an iterative fashion, re-weighting the observations for the next split depending on the values of the residuals from the previous split.

  Essentially, a combination of "weak" trees that together provide a stronger prediction.

- Bagging - fit many trees, with each tree using a bootstrap sample of the training data.

  – Final predictions for an observation are simply the average prediction from each tree.

- Methods that combine/average predictions from a group of simpler models are called "ensemble methods."

(Groups) Why might ensemble-based approaches provide better (more accurate) predictions when compared to a single regression tree?

4 Random Forest

A clever ensemble based method that was created by Leo Breiman and USU's own Adele Cutler.

An extension of bagging that, in addition to taking bootstrap samples of the original data for each tree, also only considers a random subset of the variables when deciding how to split the tree at each node.

- The random sub-setting of the variables helps differentiate the trees, which further reduces the variance of the predictions.

SAS:

```
proc hpforest data=<dataset> seed=<random seed> scoreprole=oob;
input <all my explanatory variables>
target <my response variable>;
ods output <outputs you want printed to screen>
run;
```

4.0.1 Model Accuracy

- Bootstrap samples are samples with replacement from the original data, which means some observations show up more than once in each sample, and other observations do not show up at all.

- This means that each observation will have been ignored when creating some subset of the trees.

- We can determine the out of bag (OOB) error rate by making predictions using only the trees from which a particular observation was not included in the fitting.

4.0.2 Variable Importance

Random Forest includes a powerful measure of variable importance:

- For each tree, look at the OOB and random permute (scramble) the values of a single predictor variable X_j.
- Pass the OOB data with the scrambled X_j information down the tree - obtain the OOB error rate.
- Compare this error with the OOB error obtained when X_j was not scrambled.
- The worse the error rate is with the scrambled X_j information, the more important X_j is to the model.

4.0.3 Limitations

- Random forests is an extremely powerful method, but is often referred to as a "black box" algorithm because it does not produce a model.
- The lack of model makes random forest more difficult to interpret.
- Random forests does offer **partial dependence plots**, which visualize the effect of each predictor holding all others constant, but these are not implemented in SAS.
- Alternatively, one can git a **generalized additive model** to try and visualize the effect of each predictor.

$$Y_i = s_0 + s_1\left(X_{i,1}\right) + \cdots + s_{p-1}\left(X_{i,p-1}\right) + \epsilon_i$$

5 Helpful Resources

(both from USU's Dr. Richard Cutler):

- "What Statisticians Should Know about Machine Learning"' (2017 SAS Global Forum proceedings) `https://support.sas.com/resources/papers/proceedings17/0883-2017.pdf`
- "Prediction and Interpretation for Machine Learning Regression Methods" (2018 SAS Global Forum proceedings) `https://pdfs.semanticscholar.org/eade/6d9e5a9e5e3667cb2f88c665638735c8pdf`

Remember, the less our model *assumes*, the more data we must *consume*.