4.1: Penalized Regression

Dr. Bean - Stat 5100

1 Why Penalized Regression?

Recall linear regression model and predictive equation:

$$\begin{aligned} Y &= \beta_0 + \beta_1 X_1 + \ldots + \beta_{p-1} X_{p-1} + \varepsilon \\ \hat{Y} &= b_0 + b_1 X_1 + \ldots + b_{p-1} X_{p-1} \end{aligned}$$

IF the assumptions regarding residuals are satisfied, then ordinary least squares (OLS) provides the best (i.e. minimum variance) unbiased estimator for each $\beta_k \, (k = 1, \ldots, p-1)$ using the **loss function**

$$\sum_{i=1}^{n} \left(Y_i - \hat{Y}_i \right)^2.$$

However, when multicollinearity is present, the variance of the estimates for the β_k are inflated. What we would like is a way to shrink the variance of our estimated coefficients, perhaps forcing some coefficients all the way to zero (i.e. variable selection). This will allow us to **stabilize** our coefficient estimates while at the same time provide an alternative approach for variable selection.

However, nothing in statistics comes free. Like the "soul stone" from the avengers series, we must sacrifice something we love in order to obtain smaller variance and a new approach for variable selection.

Our Solution: Sacrifice **unbiased** estimates of the β coefficients in order to reduce their variance.

(Individual) What does it mean to be unbiased?

$$E(b_k) = \beta_k$$

In other words, if I were to use multiple *different* samples to fit my regression line, the estimated coefficients will all be different, but will all be centered around the true (and unknown) coefficients. This is important because it means that as my sample size increases, I expect to get estimates that are closer and closer to the "truth".

(Why might we be OK with giving up unbiasedness in order to minimize variance?

- Coefficients are biased to have smaller magnitude compared to the "truth" so we can still interpret the sign of each estimator.
- Biased, yet stable, estimates of the coefficients can often provide greater predictive accuracy than an OLS model.

2 Penalized Regression Approaches

Alternative Loss Functions:

- Ridge regression

$$\sum_{i=1}^{n}\left(Y_i - \hat{Y}_i\right)^2 \quad + \quad \lambda \sum_{k=0}^{p-1}(\beta_k)^2$$

- LASSO

$$\sum_{i=1}^{n}\left(Y_i - \hat{Y}_i\right)^2 + \lambda \sum_{k=1}^{p-1}|\beta_k|$$

- Adaptive LASSO

$$\sum_{i=1}^{n}\left(Y_i - \hat{Y}_i\right)^2 + \lambda \sum_{k=1}^{p-1}\frac{|\beta_k|}{\tilde{b}_k}$$

  - Where $\tilde{b}_k$ represents some initial estimate of the model coefficients (perhaps using OLS or traditional LASSO).

- Elastic Net

$$\sum_{i=1}^{n}\left(Y_i - \hat{Y}_i\right)^2 + \lambda_1 \sum_{k=0}^{p-1}(\beta_k)^2 + \lambda_2 \sum_{k=1}^{p-1}|\beta_k|$$

- Select values of λ that balances added bias with reduced variance.
- Our goal is impose the least amount of biasedness that we can in order to achieve an acceptable reduction in variance.
- One potential solution would be to select λ in such a way that minimizes the cross validation error.

Check out https://ww2.amstat.org/meetings/csp/2014/onlineprogram/handouts/T3-Handouts.pdf for additional info on these approaches.

Note that the explanatory variables MUST be standardized in order to use penalized regression techniques. Many functions perform this standardization automatically "under the hood."

2.1 Ridge Regression

Recall Linear Algebra Representation of OLS Regression:

$$Y = X\beta + \epsilon b \qquad = (X'X)^{-1}X'Yb \sim N(\beta, (X'X)^{-1}\sigma^2)$$

Recall also how we can standardize our X and Y variables producing:

$$\begin{aligned} Y^* &= X^*\beta^* + \varepsilon \\ b^* &= (X^{*\prime}X^*)^{-1}X^{*\prime}Y^* \\ &= (r_{XX})^{-1}r_{YX} \\ Cov(b^*) &= (r_{XX})^{-1}\sigma^2 \end{aligned}$$

$$\begin{aligned} Y_i^* &= \frac{1}{\sqrt{n-1}} \cdot \frac{Y_i - \bar{Y}}{\text{SD of } Y} \\ X_{k,i}^* &= \frac{1}{\sqrt{n-1}} \cdot \frac{X_{k,i} - \bar{X}_k}{\text{SD of } X_k} \\ r_{XX} &= \text{correlation matrix of } X\text{'s} \\ r_{YX} &= \text{correlation vector between } Y \text{ and } X\text{'s} \end{aligned}$$

Ridge Regression introduces a small positive biasing constant $\lambda > 0$ so that

$$b^R = (r_{XX} + \lambda \cdot I)^{-1}r_{YX}$$

where I is the identity matrix (one's on the diagonal of the matrix and zeros elsewhere).

SAS Code:

```
proc reg data=<dataset> ridge=0 to <upper bound> by <step size>
     outvif outest=<named dataset of relevant ridge output>
     plots(only)=ridge(VIFaxis=log);
  model <model statement> / vif;
run;
```

Two graphical summaries to choose the "right" ridge parameter c:
(Note: these are guides; there is no "optimal" decision)

1. Ridge Trace Plot
   - (Need standardized data for this to be meaningful; SAS does internally)
   - Simultaneous plot of $b_1^R, \ldots, b_{p-1}^R$ (using standardized data) for different ridge parameters c (usually from 0 to 1 or 2)
   - As c increases from 0, the b_k^R may fluctuate wildly and even change signs
   - Eventually the b_k^R will move slowly toward 0
2. VIF Plot
   - Simultaneous plot of the variance inflation factor for the $p-1$ predictors for different ridge parameters

- As c increases from 0, the VIF drop toward 0

In general, choose smallest ridge parameter c:

1. where the b_k^R first become "stable" (their approach towards 0 has slowed)
2. and the VIF's have become "small enough" (close to 1 or less than 1)

2.1.1 Comments on Ridge Regression

- Choice of ridge parameter is somewhat subjective, but must be defendable (i.e. with a trace plot)
- given ridge parameter c, can get resulting parameter estimates b on the "unstandardized" (original data) scale
  - SAS gives these automatically, but need textbook equation 7.46b to get intercept b_0:

$$\beta_0 = \bar{Y} - \beta_1\bar{X}_1 - \beta_2\bar{X}_2 - \ldots - \beta_{p-1}\bar{X}_{p-1}$$

- ridge regression estimates b tend to be more robust against small changes to data than are OLS estimates
- predictors with very unstable ridge trace (tends toward zero without any plateau or slowing down) may be dropped from model, providing an alternative to stepwise variable selection techniques
- **major limitation**: traditional inference is not directly applicable to ridge regression estimates (part of our "soul stone" sacrifice)

2.2 LASSO (Least Absolute Shrinkage and Selection Operator)

Find b to minimize

$$\sum_{i=1}^{n}\left(Y_i - \hat{Y}_i\right)^2 + \lambda\sum_{k=1}^{p-1}|\beta_k|$$

Switching from $\lambda\sum_{k=1}^{p-1}\beta_k^2$ in ridge regression to $\lambda\sum_{k=1}^{p-1}|\beta_k|$ in LASSO, may seem minor, but this change causes b_k values to now shrink all the way to zero.

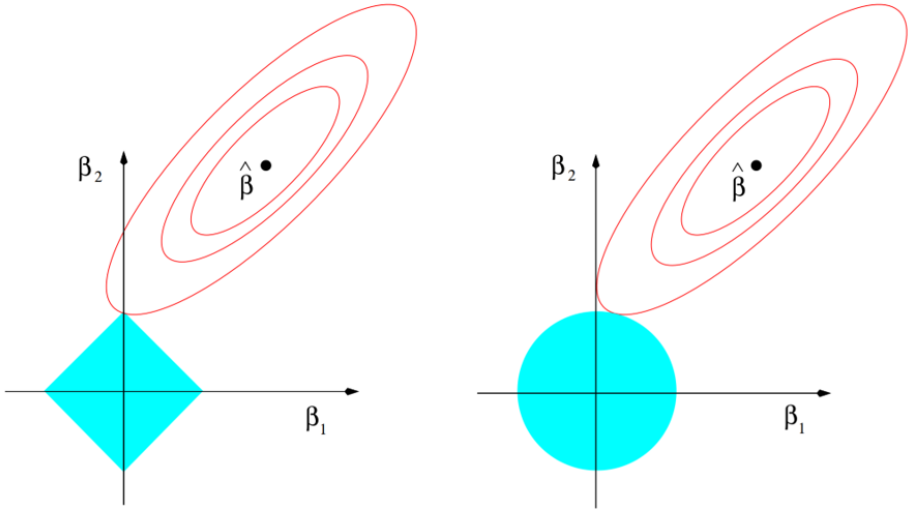

FIGURE 3.11. *Estimation picture for the lasso (left) and ridge regression (right). Shown are contours of the error and constraint functions. The solid blue areas are the constraint regions* $|\beta_1| + |\beta_2| \leq t$ *and* $\beta_1^2 + \beta_2^2 \leq t^2$*, respectively, while the red ellipses are the contours of the least squares error function.*

Options exist for choosing λ We can use these because we now have models with different numbers of coefficients, not the case in ridge regression.

- likelihood function-based criteria (Adj. R^2, C_p, AIC, SBC, etc.)
- cross-validation
  - withhold some of the data, fit on the rest, then predict on withheld portion
  - select λ to minimize something like (others exist)

$$PRESS = \sum_{i=1}^{n} \left(\frac{Y_i - \hat{Y}_i}{1 - h_{ii}} \right)^2$$

SAS Code

```
proc glmselect data=<dataset> plots=(criterion <measure>);
 class <all qualitative variables>;
 model <your model>
        / selection=lasso(adaptive choose=<selection method> stop=none);
 output out=<output dataset> p=<lasso predictions>;
run;
```

One way to visualize progress of model is to show ASE as each variable is added

$$\text{ASE} = \frac{\text{SSE}}{n} \qquad MSE = \frac{\text{SSE}}{n-p}$$

.

2.3 Adaptive LASSO

- Problem: LASSO is known to give more biased estimates of nonzero coefficients
- Solution: Allow higher penalty for zero coefficients and lower penalty for nonzero coefficients

Find b to minimize

$$\sum_{i=1}^{n}\left(Y_i - \hat{Y}_i\right)^2 + \lambda \sum_{k=1}^{p-1} \frac{|\beta_k|}{b_k}$$

"Adaptive" weights: $\frac{1}{b_k}$, where b_k is obtained from an initial model fit (using OLS or regular LASSO or something else)
– control shrinking of zero coefficients more than nonzero coefficients

2.4 Elastic Net

> *Similar to the lasso, the elastic net simultaneously does automatic variable selection and continuous shrinkage, and it can select groups of correlated variables. It is like a stretchable fishing net that retains 'all the big fish'"* - Zou and Hastie (2005)

Some limitations of LASSO:

- When number of predictors $(p-1)$ exceeds sample size (n), LASSO will select up to n predictor variables before it saturates.
- In the presence of high multicollinearity, LASSO tends to select only one variable from the group of correlated predictors.
- When sample size (n) exceeds number of predictors $(p-1)$ and there is high multicollinearity, LASSO is out-performed (prediction-wise) by ridge regression.

Elastic Net overcomes these limitations:

- can select more than n variables
- can select more than one variable from a group of highly collinear predictors
- can achieve better predictive performance

Find b to minimize

$$\sum_{i=1}^{n}\left(Y_i - \hat{Y}_i\right)^2 + \lambda_1 \sum_{k=0}^{p-1} (\beta_k)^2 + \lambda_2 \sum_{k=1}^{p-1} |\beta_k|$$

4.2: Variations on OLS (Ordinary Least Squares)

Dr. Bean - Stat 5100

1 Why alternatives?

Remember this: when standard model assumptions are met, OLS is the "best" linear modeling approach.
No matter how good we are at performing variable transformations, there are some situations where we simply cannot satisfy linear model assumptions of constant variance, normality, or independence.

Fortunately, there are several OLS alternatives that address one or more of these issues.

The cost:

- Lose our ability to conduct inference on the coefficients.
- The models become harder to fit/harder to explain.

2 Weighted Least Squares (textbook §11.1)

Recall regression model $Y = \beta_0 + \beta_1 X_1 + \ldots + \beta_{p-1} X_{p-1} + \varepsilon$ in matrix form:
(Ch. 5, Handout #12 p. 2)

$$Y = X\beta + \varepsilon$$

Model assumption: $\varepsilon \sim N(0, \sigma^2 I)$

- If constant variance, (i.e., $Cov(\varepsilon) = \sigma^2 I$), then use OLS:

$$b = (X'X)^{-1}X'Y$$

- If non-constant variance, then can estimate and account for it (WLS):

$$V = Cov(\varepsilon) \qquad \text{(typically assumed diagonal)}$$

$$W = V^{-1} \qquad \text{(i.e. the weights)}$$

$$b_w = (X'WX)^{-1}X'WY$$

Why give *smaller* weight to observations with *larger* variance when calculating the model coefficients?

Smaller variance is equivalent to greater certainty. Certain information should have greater "value" than uncertain information.

Typically, $Cov(\varepsilon)$ must be estimated

- can often relate variance of residuals (or squared residuals) to predictors or $\hat{Y}$ values
- example (as in Ex. 1 of Handout 4.2.1): residual vs. X_1 is megaphone-shaped (linear relationship between SD of residual and X_1)
  - regress absolute residuals on X_1 and get predicted values s (as function of X_1)
  - define weights $w = 1/s^2$
- see p. 425 for other examples
  – key is how to estimate w for given scenario, as a function of X's

Some things to remember:

- The *pattern* of the residuals against the other variables determines how we should estimate the weights.
- Its OK to see non-constant variance in weighted model.
- In **Spatial Statistics, weights are calculated using geographic similarity.**

3 Robust Regression (textbook §11.3)

Rather than remove influential observations and outliers, we may choose to reduce their influence by changing the way we measure "error".

3.1 IRLS (iteratively reweighted least squares)

1. Obtain (maybe from OLS) b, then calculate $\hat{Y} = Xb$ and $e = Y - \hat{Y}$
2. Calculate weights W, based on e (lots of weight functions available)
3. Calculate (WLS) $b_w = (X'WX)^{-1}X'WY$ and resulting $e = Y - Xb_w$
4. Iterate steps 2 & 3 to convergence of b_w

How to calculate weights?
- usually chosen to optimize some criterion
- the choice of criterion determines the method of weight calculation

3.2 M-estimation

- If $u_1, \ldots, u_n$ are *iid* from some distribution with parameter θ, then the type-M estimate of θ is of the form

$$\hat{\theta} \;=\; \arg\min \sum \rho(u_i; \theta)$$

  where ρ is some "scalar objective function"

- Example: $\rho(u;\theta) = -\frac{1}{n}\log f(u;\theta)$, f is pdf of distribution of $u_1, \ldots, u_n$. Then

$$\begin{aligned} \hat{\theta} &= \arg\max \sum \frac{1}{n} \log f(u_i;\theta) \\ &= \arg\max (\text{likelihood}) \\ &= (\text{what is this called?}) \end{aligned}$$

- W-estimation approach in IRLS:
  1. Calculate robust estimate of σ, such as $s = \frac{MAD(e)}{0.6745}$
  2. Let $u_i = \frac{e_i}{s}$ be "scaled" (or standardized) residual
  3. Calculate (diagonal) weights $w_i = \frac{\psi(u_i)}{u_i}$
     – where $\psi(u) = \rho'(u)$ for some scalar objective function ρ

Example – Tukey Bisquare (sometimes called Tukey's Biweight):

$$\rho(u) = \begin{cases} \frac{c^2}{3}\left(1 - [1 - \left(\frac{u}{c}\right)^2]^3\right) & \text{if } |u| \le c\text{; default } c = 4.685 \\ \frac{c^2}{3} & \text{otherwise} \end{cases}$$

Bisquare weight function: $w(u) = \left(1 - \left(\frac{u}{c}\right)^2\right)^2$ for $|u| \le c$, 0 otherwise

Note: M-estimation works well for outliers; for leverage points, use MM-estimation (see SAS help)

3. Nonlinear Regression (textbook §13.1 – 13.2)

What if Y vs $X_1, \ldots X_{p-1}$ not linear (in β's)?
– Usually need mechanistic theory

Mechanistic Theory: the assumption that a natural phenomenon can be understood through the use of an equation.
Example: Population Growth

$$\frac{dN}{dt} = rN(1 - N/K)$$

`proc nlin` fits these nonlinear models

- Parameters estimated by an iterative process to reduce the SSE at each iteration, until convergence
- Keys to [useful] convergence:
  - form of nonlinear equation
  - initial parameter estimates

If you were dropped randomly on the side of a mountain with dense fog, how would you find your way down? How would you know when you have made it to the bottom (assuming the fog persists at the bottom)?

You would most likely take each step in a direction that would cause you to be lower than you were before. You would know that you (hopefully) arrived at the bottom of the mountain when you can no longer find a direction to take a step in which you could decrease your altitude. This approach is often called **gradient descent**.

Example 3.1: $Y = \beta_0 + \beta_1 X_1^{\beta_2} - \beta_3 \exp(\beta_4 X_2) \quad (+\epsilon)$
(with simulated data)

Example 3.2: a nonlinear curve to describe sand compression, from Lagioia et al. (1996) Computers and Geotechnics 19(3):171-191

$$f = \frac{p}{p_c} - \frac{\left(1 + \frac{q}{p \cdot M \cdot k_2}\right)^{\frac{k_2}{(1-\mu)(k_1-k_2)}}}{\left(1 + \frac{q}{p \cdot M \cdot k_1}\right)^{\frac{k_1}{(1-\mu)(k_1-k_2)}}},$$

where

f = yield surface (response)
q = deviatoric stress (predictor)
p = mean effective stress (predictor)
p_c = hardening / softening constant defining current size of surface (known)
η = stress ratio p/q
M = parameter defining value of η with no strain increment
μ = parameter defining general slope of d vs. η curve
α = parameter defining how close to $\eta = 0$ axis curve bends towards $d = \infty$
d = dilatancy, $2\mu M(1 - \alpha)$

Goal: find μ, α, and M to make $f \approx 0$, and look at the relationship between these three parameters

`proc model` estimates such nonlinear systems (can do multiple equations)

From playing with this in SAS, it appears that to achieve convergence of estimates in `proc model`, the most important thing is that at least one of the tails of the $q * p$ curve to be fit has data along most of it. To make the convergent estimates "good", it appears necessary to have data along both tails. It is also crucial that the initial starting estimates be good, especially for M (maybe within .2 or so).

4.3: 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.

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

- If residual distributional assumptions are violated, there can be no meaningful model inference.
- Our accuracy will likely be poor if we assume the wrong model form.

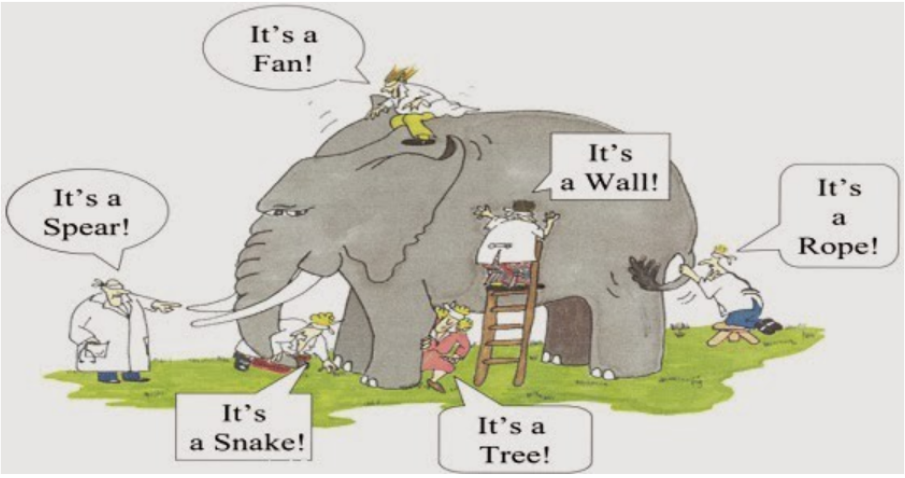

Figure 1: `https://medium.com/betterism/the-blind-men-and-the-elephant-596ec8a72a7d`

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"

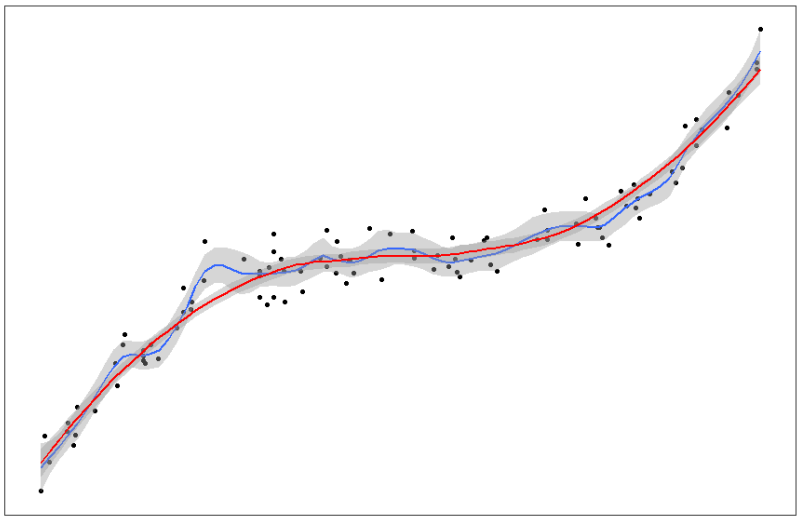

Figure 2: Example LOESS smoothing curves with only one X-variable and two levels of smoothness.

- 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** ($\sum_i \left(Y_i - \hat{Y}_i\right)^2 = 0$).
  - 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.

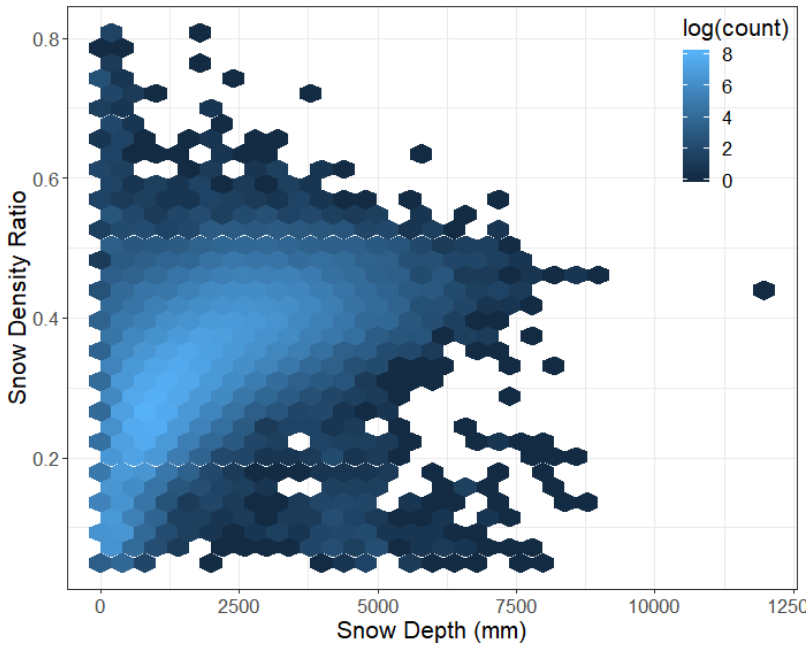

Figure 3: Plot of the snow density ratio in relation to its depth for locations across North America.

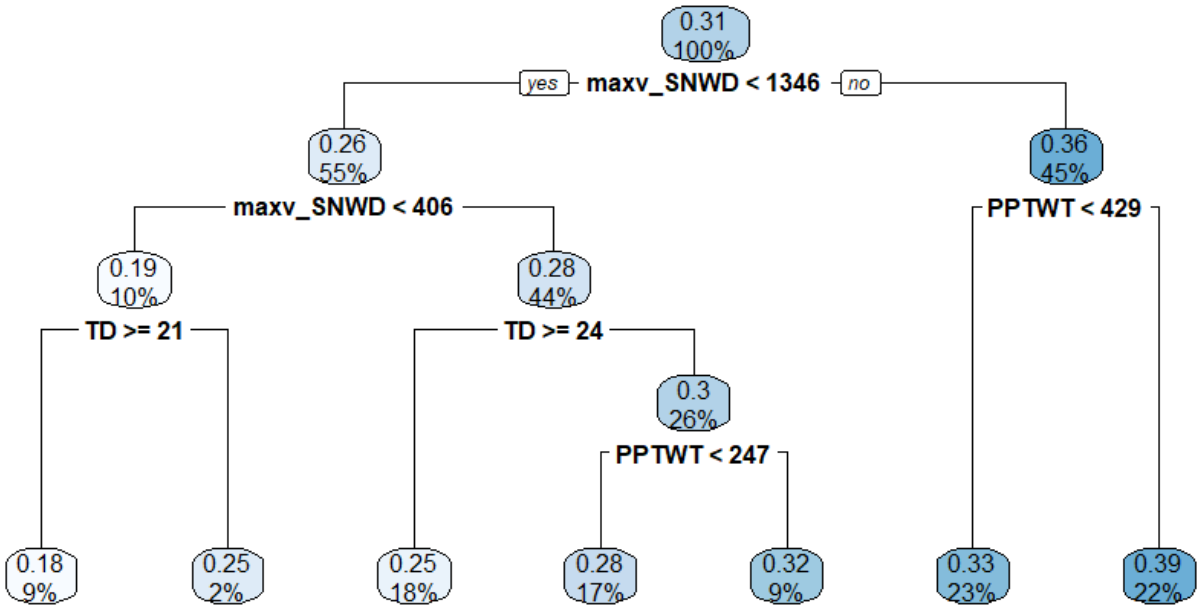

Figure 4: Sample tree (pruned) for predicting snow ratio using climate variables.

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."

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

Taking the **average** of a set of predictions has the effect of reducing the **variance** of the overall prediction. Reductions in variance lead to an overall increase in accuracy.

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 get 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*.