Thick Data in R

Alexander C. Mueller, PhD

November 6, 2019

Agenda

- Why understanding thick data is important
  - math intuition
  - practical concerns
- Regularization
  - Ridge and LASSO
  - how and why they work
- Trees and Forests
  - decision trees as implicit feature selection
  - bootstrapping and tree parameters

Thick Data First Notions

Thick data has **many predictors and relatively few observations**. It is not a precise notion, but a heuristic that any modeling algorithm may give you trouble if you have too many columns relative to not enough rows.

Algorithms may misbehave for different reasons, but most relate to how high-dimensional space is **locally bigger** and this mathematical fact is actually pretty powerful. This is called the **"curse of dimensionality"** in some sources.

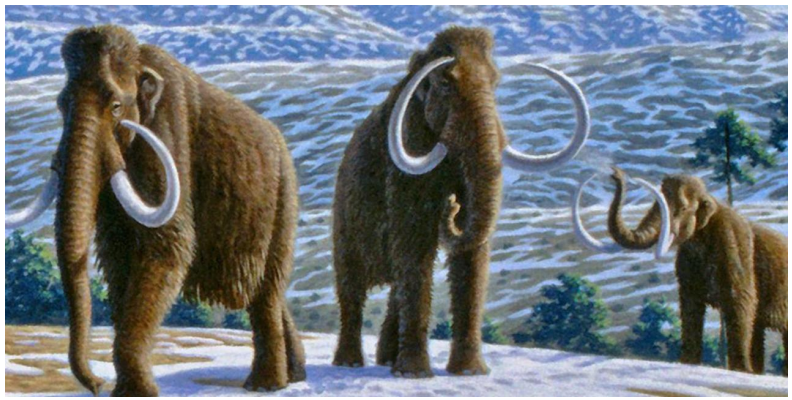
Excess error from thickness is probably **variance** and not bias - your model has been given unusual opportunity to **memorize noise** in the data.

Discussion Question

What would your life be like in 1000 spatial dimensions?

Lighter Fare: Who's Hot, Who's Cold?

Size Keeps You Warm



- Generate heat in proportion to your volume.
- ... a cube of side-length r has volume r^3.
- Radiate heat in proportion to your surface area.
- ... a cube of side-length r has surface area $6r^2$.

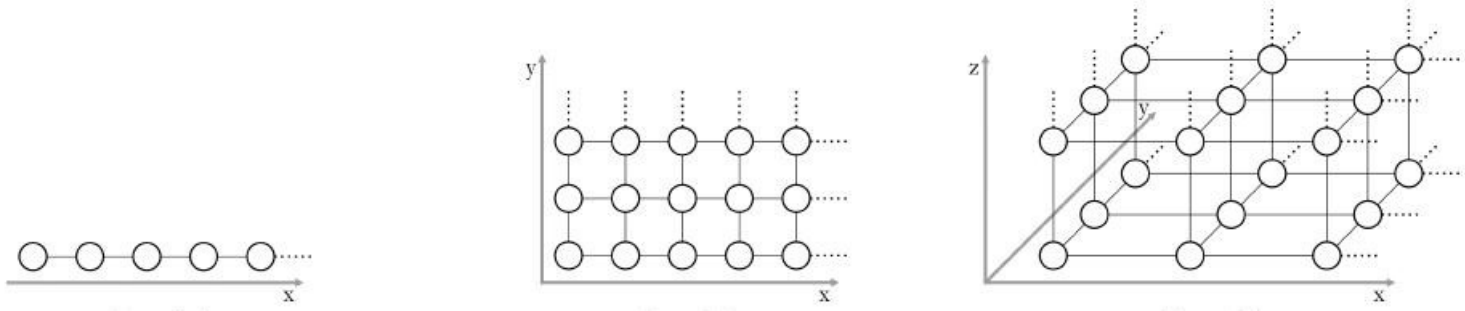
$$\text{StV} = \frac{\text{surface area}}{\text{volume}} = \frac{6r^2}{r^3} = \frac{6}{r} \to 0 \text{ as } r \to \infty, \ \to \infty \text{ as } r \to 0$$

Keeping Warm in High and Low Dimensions

The arithmetic of surface area vs. volume changes in higher dimensions. For an n cube of radius r...

- $n = 1$ (a line segment) $\Rightarrow$ StV $= \frac{2}{r}$
- $n = 2$ (a square) $\Rightarrow$ StV $= \frac{4}{r}$
- $n = 3$ (an honest cube) $\Rightarrow$ StV $= \frac{6}{r}$
- ... $\Rightarrow$ StV $= \frac{2n}{r}$

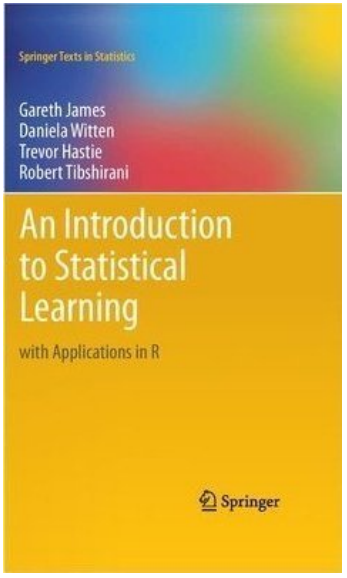
Recurrence for Random Walks



Wander randomly. Do you make it back where you started?

- In any dimension, you either find your way back again and again (recurrence) or eventually become totally lost forever.
- For dimension one or two, you return to your starting point infinitely many times with probability one.
- For dimension more than three, you eventually wander off never to return with probability one.
- High dimensional space has more “extra space” locally.

Principal Reference



The following borrows some material from Chapter 6 of "An Introduction to Statistical Learning" by Hastie, Tibshirani, et al. and the book is available free online. (link).

The R code in the Chapter 6 lab is especially relevant.

Slides and code at github.com/capnion/random

Where to find thick data

A non-exhaustive list of situations that often produce thick data:

- **Categorical data** often creates it, especially if a variable with many levels is encoded "one-hot" (perhaps not by your direct decision).
- **Bureaucracy** tends to create it, often when inclusion of an administrative code creates a categorical variable with many levels.
- Going out and getting data with **more predictors** on your existing dataset is **popular and often helpful**, but eventually gives you a thick data problem.

Oops, it's thick now

```
fm1 <- lm(
   Income ~ state.division + HSGrad,
   data = stateFrame
)
summary(fm1)
```

```
Coefficients:
                                 Estimate Std. Error t value Pr(>|t|)
(Intercept)                       1175.12     844.02   1.392 0.171530
state.divisionMiddle Atlantic      630.96     302.78   2.084 0.043601
state.divisionSouth Atlantic       481.72     267.72   1.799 0.079514
state.divisionEast South Central   -10.28     348.02  -0.030 0.976574
state.divisionWest South Central   -73.47     309.73  -0.237 0.813703
state.divisionEast North Central   354.34     257.40   1.377 0.176285
```

What goes wrong

Thick data can create a variety of issues for a model...

- **Poor model performance** - unnecessary variables add variance and increase error.
- Inevitable **multicollinearity** complicates things...
  - Whether a variable is significant may depend on which other variables are included.
  - Various metrics may suggest excluding a variable that is clearly important.
- **Inference can be unwise** or difficult on thick data.

Thick data tends to create big headaches for model interpretation because of uncertainty around how a model should be specified.

Inference Concerns

Thick data can cloud inference about which predictors are important.

Obsession with *p*-values is an example of how sometimes this is all anyone cares about.

```
Coefficients:
                                Estimate Std. Error t value Pr(>|t|)
(Intercept)                      1175.12     844.02   1.392 0.171530
state.divisionMiddle Atlantic     630.96     302.78   2.084 0.043601
state.divisionSouth Atlantic      481.72     267.72   1.799 0.079514
state.divisionEast South Central  -10.28     348.02  -0.030 0.976574
state.divisionWest South Central  -73.47     309.73  -0.237 0.813703
state.divisionEast North Central  354.34     257.40   1.377 0.176285
```

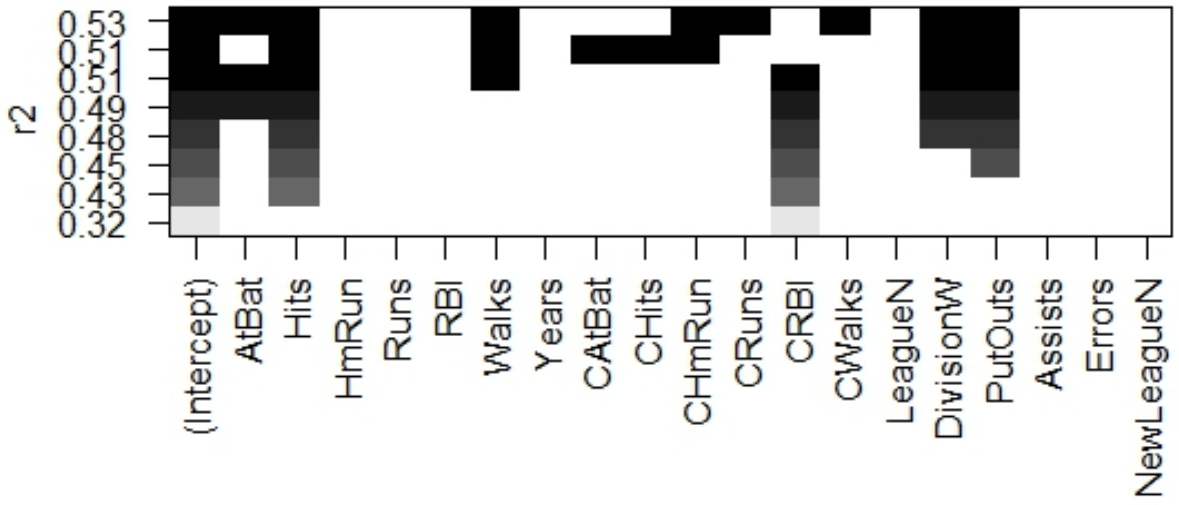
Notable Folk Wisdom

There is a vein of folk wisdom you could say is about knowing how to avoid thick data issues with more observations or better parameter choices.

When $p >= n$ linear regression is not even well-posed from a linear algebra standpoint, and best practice is $p < \sqrt{n}$

Split on $\sqrt{p}$ predictors in each step of growing a tree in a random forest, and this choice has consequences for other parameters.

Mo' Variables, Mo' Problems 1



This graph (from the lab) shows the R^2 of a sequnce of models, each the optimal model of 1, 2, 3, ... variables selecting using the best selection approach. Is CRBI important or not? AtBat?

Mo' Variables, Mo' Problems 2

```
> #best model of 5 variables
> lm(Salary~Hits+CRBI+Division+PutOuts,data=Hitters)

Call:
lm(formula = Salary ~ Hits + CRBI + Division + PutOuts, data = Hitters)

Coefficients:
(Intercept)         Hits         CRBI    DivisionW      PutOuts
    13.9231       2.6758       0.6818    -139.9539       0.2735
```

Mo' Variables, Mo' Problems 3

```
> #best model of 6 variables
> lm(Salary~Hits+AtBat+CRBI+Division+PutOuts,data=Hitters)

Call:
lm(formula = Salary ~ Hits + AtBat + CRBI + Division + PutOuts,
    data = Hitters)

Coefficients:
(Intercept)         Hits        AtBat         CRBI    DivisionW      PutOuts
    97.7684       7.1753      -1.4401       0.6882    -129.7319       0.2905
```

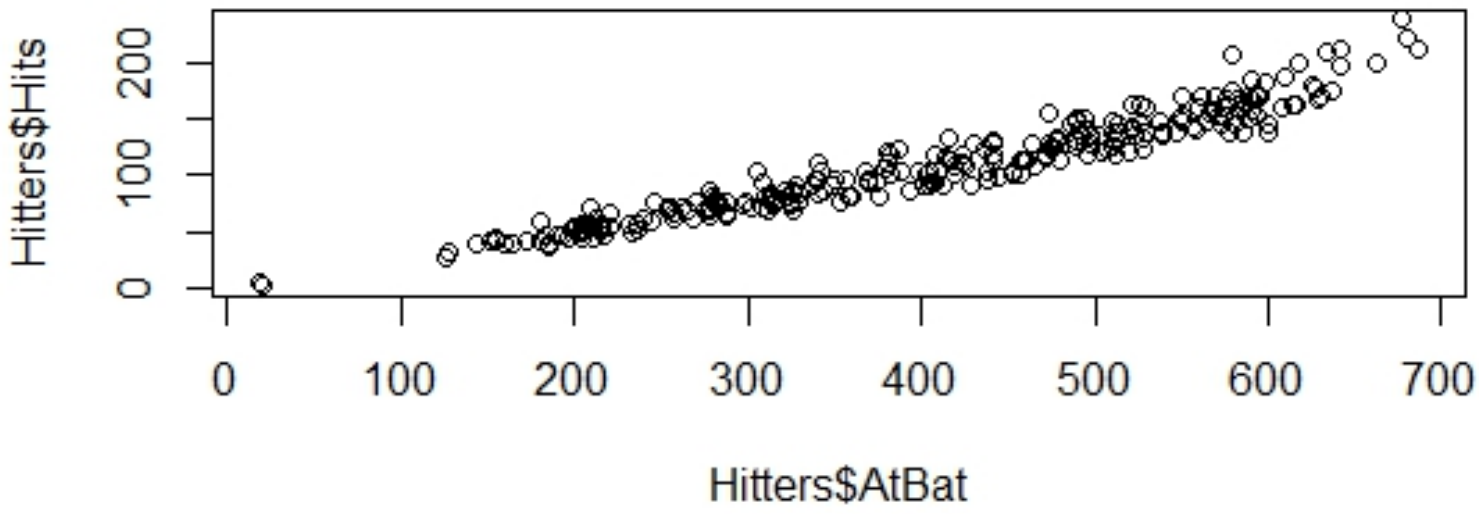
Mo' Variables, Mo' Problems 4

```
> #the model of 5 variables with atbats in place of hits
> lm(Salary~AtBat+CRBI+Division+PutOuts,data=Hitters)

Call:
lm(formula = Salary ~ AtBat + CRBI + Division + PutOuts, data = Hitters)

Coefficients:
(Intercept)        AtBat         CRBI    DivisionW      PutOuts
    26.3518       0.6716       0.6942    -148.4700       0.2916
```

Mo' Variables, Mo' Problems 5



Mo' Variables, Mo' Problems 6

The instability of the AtBat variable is a common problem that can dangerously undermine some conventional approaches to analyzing a model.

- Whether we accept a hypothesis about a coefficient can depend on our subsetting choices.
- The potential reversal of a coefficient's sign is a fatal problem for interpretation.
- Instability in coefficients can drive variance in predictions.
- Multicollinearity in thick data can be unavoidable.

We *must* make some tough decisions about what to include in our model.

Shrinkage Methods

Shrinkage methods estimate the coefficients under additional constraints (or *regularizes* them) and in practice this is to make them smaller. The model is penalized not only for error but for having "too many" coefficients, with different penalties giving different behaviors...

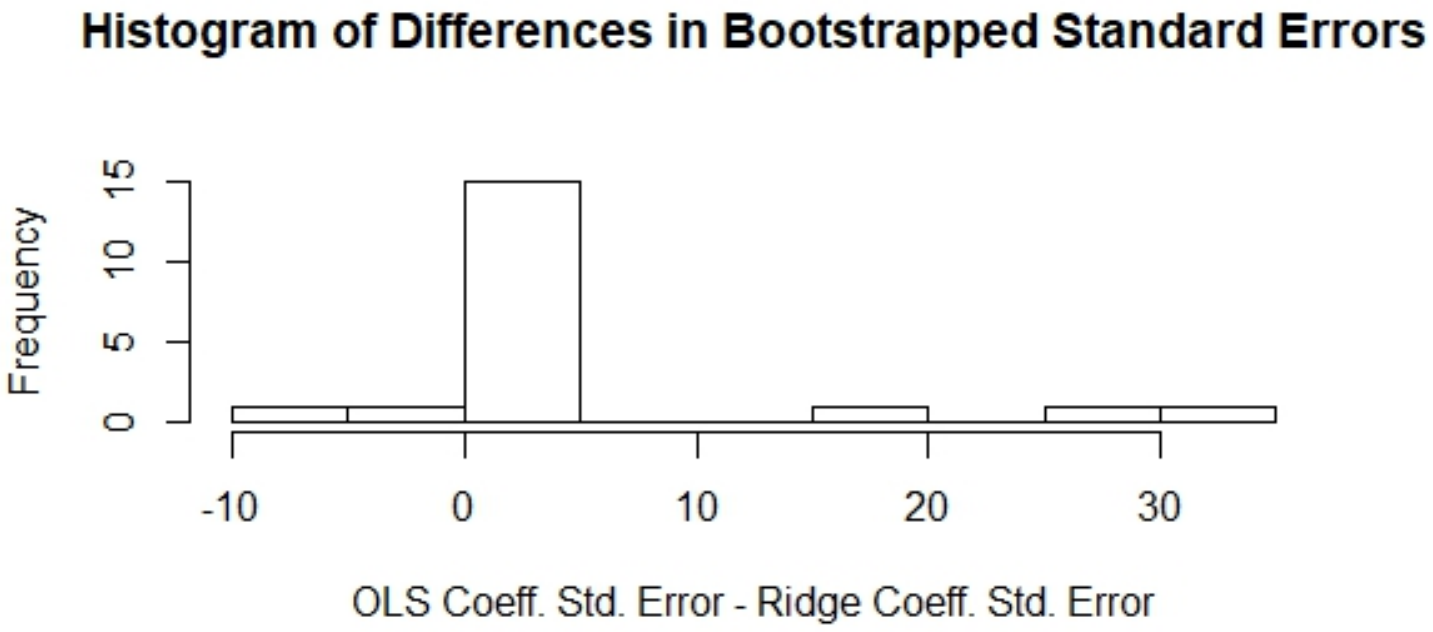
- Ridge Regression: shrinks coefficients, uses L^2 norm.
- Lasso: shrinks some coefficients to 0, uses L^1 norm.

Might or might not be the best model period, and also clarifies which predictors are important.

Lasso Lab Example From Lab

```
> lasso.coef
 (Intercept)        AtBat         Hits        HmRun         Runs          RBI
  18.5394844    0.0000000    1.8735390    0.0000000    0.0000000    0.0000000
       Walks        Years       CAtBat        CHits       CHmRun        CRuns
   2.2178444    0.0000000    0.0000000    0.0000000    0.0000000    0.2071252
        CRBI       CWalks      LeagueN    DivisionW      PutOuts      Assists
   0.4130132    0.0000000    3.2666677 -103.4845458    0.2204284    0.0000000
      Errors   NewLeagueN
   0.0000000    0.0000000
```

Ridge vs. OLS Standard Errors



Ridge pushes many of the smaller coefficients close to zero.

Code and a Wrinkle 1

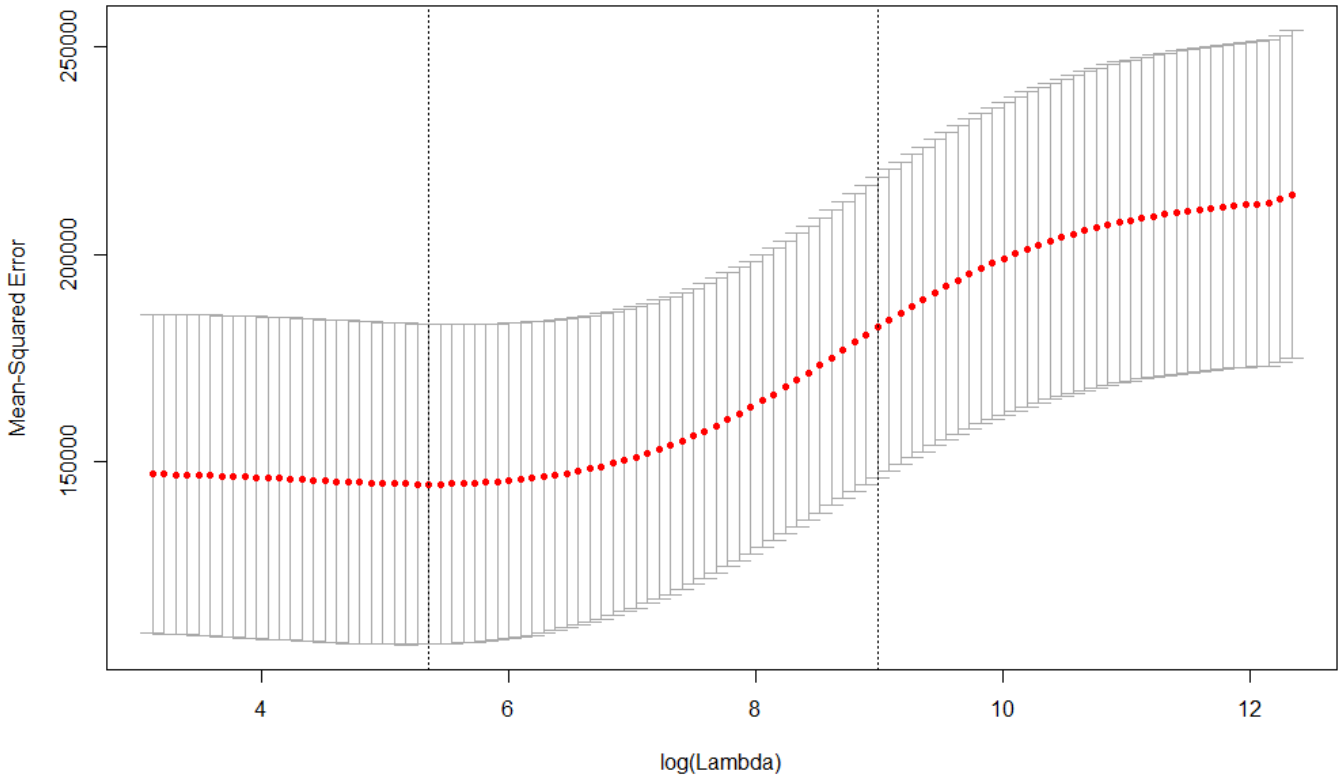
```
x=model.matrix(Salary~.,Hitters)[,-1]
y=Hitters$Salary

# Ridge Regression

library(glmnet)
grid=10^seq(10,-2,length=100)
ridge.mod=glmnet(x,y,alpha=0,lambda=grid)
```

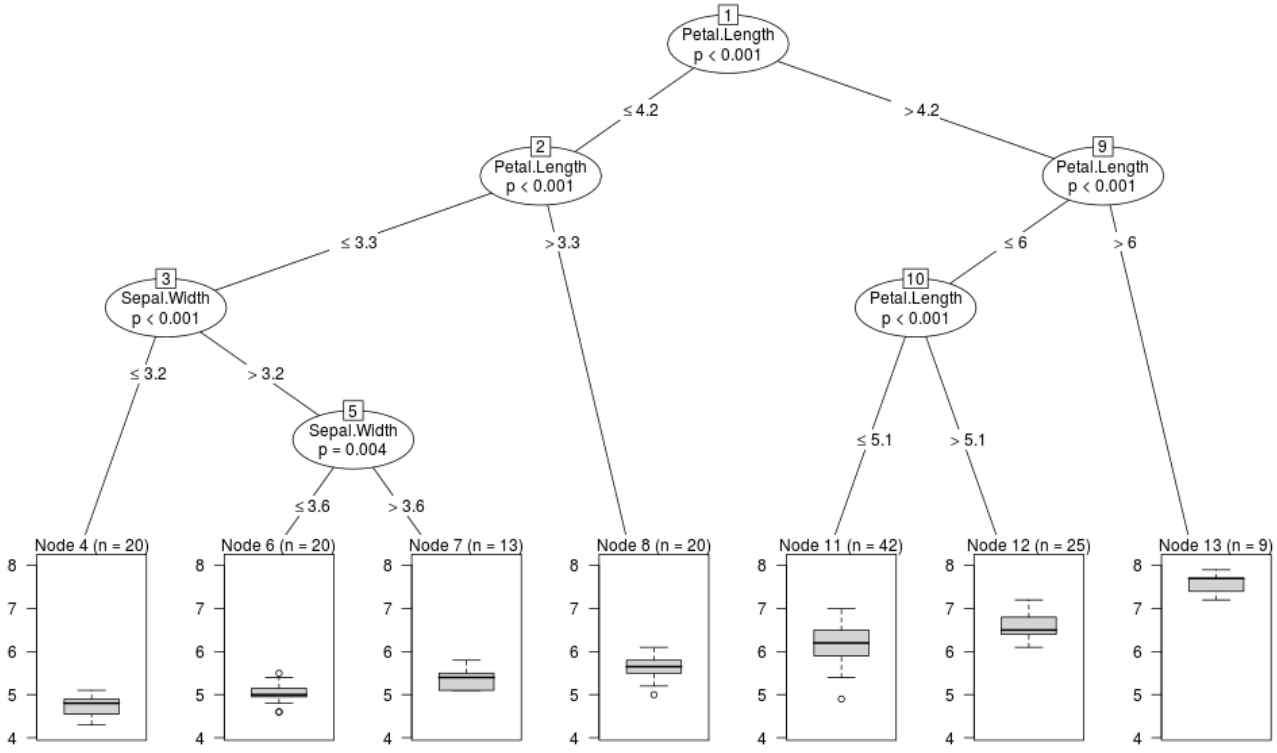
The lambda argument determines a range of values for a parameter that determines the severity of the penalty applied to large coefficients.

Code and a Wrinkle 2



One hopes that this graph has a local minimum somewhere, as it indicates the model was improved by shrinking the coefficients.

Tree Basics: Sepal Length in Iris



Decision trees are implicit feature selection, as they may entirely ignore predictors that do not produce good splits.

Common Parameters

There are a few parameters that are common to most or all algorithms that learn using multiple trees, such as...

- the number of trees used B.
- the shrinkage parameter λ, controlling how fast the algorithm tries to learn at each step.
- the number of splits d (or “mtry”) in each tree, a high-level measure of complexity.

Fine-tuning these parameters is done by cross-validation and there is little opportunity for parameter estimation.

Random Forest Under the Hood

```
> rf
Random Forest

669 samples
 12 predictor
  2 classes: 'DiCaprio', 'Winslet'

No pre-processing
Resampling: Bootstrapped (25 reps)
Summary of sample sizes: 542, 542, 542, 542, 542, 542, ...
Resampling results across tuning parameters:

  mtry  Accuracy   Kappa      Accuracy SD  Kappa SD
  2     0.8107367  0.5994673  0.02389990   0.05013360
  5     0.7984620  0.5779757  0.03371800   0.07040022
  9     0.7846143  0.5501639  0.03258991   0.06775746

Accuracy was used to select the optimal model using  the largest value.
The final value used for the model was mtry = 2.
```

Standard random forest via the caret package resamples and tries different values of a parameter related to thickness.

Questions

Any questions?