Advanced Models I

Practical Machine Learning (with R)

UC Berkeley

Fall 2015

Topics

- Review and Expectations
- Questions
- New Topics

REVIEW AND EXPECTATIONS

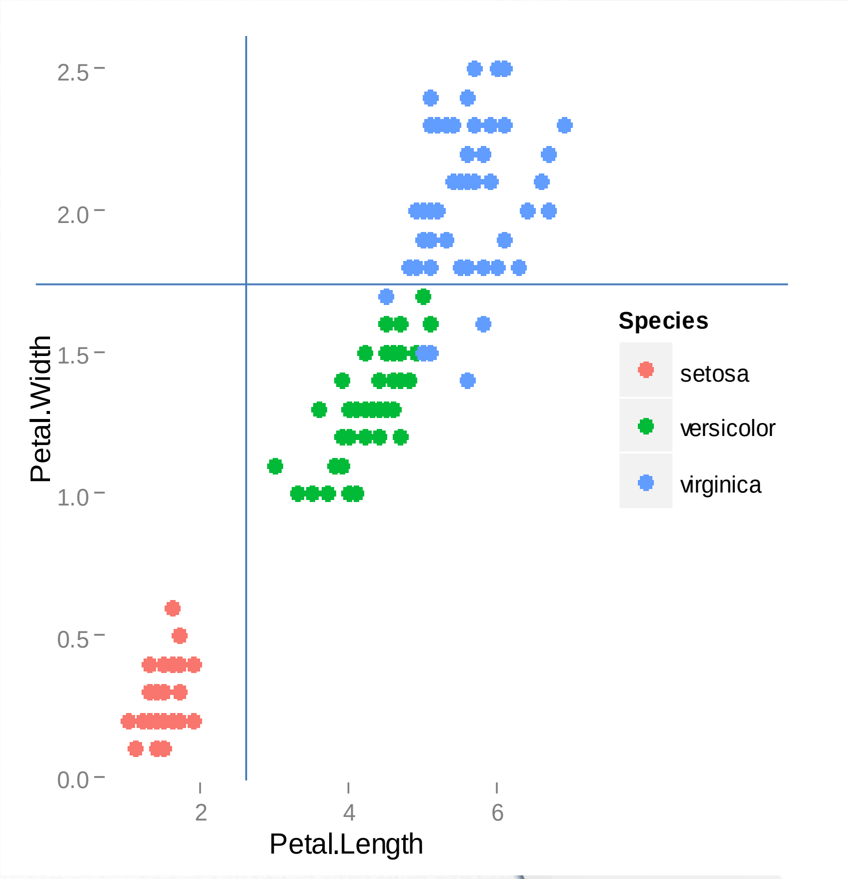
Review and Expectation

Understand **Recursive Partitioning** :

- intrinsically how recursive partitioning models work; how splits are determined; what splitting accomplishes.
- how model tuning parameters control for bias variance trade-offs.
- what `Cp` is and how to use it to prune trees to the proper size.

Trees

- All about splitting
  - Different variants have different rules for evaluating the splits
- Tree = Ruleset = Partition of Space
  - Node = Rule = “box” (contiguous region of space)



Tree Method Advantages

- List em

Tree Method Advantages

- Highly interpretable
- Easy to implement (even in SQL)
- Computationally cheap
- Handle many predictors (sparse, skewed, continuous, categorical, missing) --> little need to pre-process them
- Non-parametric: do not require specification of predictor-response relationship
- Intrinsic feature selection
- Insensitive to order preserving transformations of predictors

Review and Expectation

Use **`rpart`**:

- Build Recursive Partitioning Models using `rpart`
- Prune trees to statistically relevant size
- Plot `rpart` models using `as.party` from the **party** package

Understand Caret

- Use the **caret** package and the `train` function to build models
- Understand the difference between using `caret` and building models manually. What `caret` provides.
- Control how models are built using the `train & trainControl` functions
- How to extract the final model
- How to plot the tuning parameters

Resampling Methods

- Get more accurate estimation of a statistic/value by resampling methods

→ Generalize to more better estimation of a ***function***

QUESTIONS

NEW TOPICS

Tree Variants

- There are many tree variants

- Tweaks
  - change how splits are determined
  - when to stop growing the tree
  - how the node value is determined

Missing Data

- Missing values in predictors are common
- A split determines which observations go to the LHS and RHS. How to Handle `Nas`?
- `NA_Categorical`
  - Treat as separate category
- `NA` (in general)
  - Use **Surrogate Splits**

Surrogate Splits

- Tree is built ignoring missing data
  - Any record with incomplete data (response or predictor) is rejected -or-
  - Missing data is rejected from determined the split
- Variables are often collinear → splits are similar and send variables down the same path.
  - Choose a surrogate split that best approximates the chosen split (accuracy)
  - Very often this is also a good split.

Gini Index

- Measure node purity:

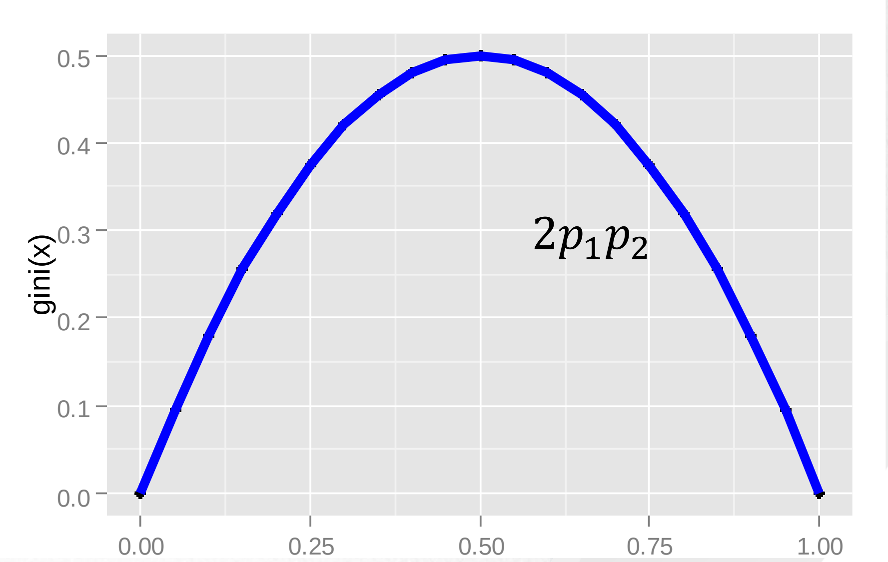
$$p_1(1 - p_1) + p_2(1 - p_2)$$

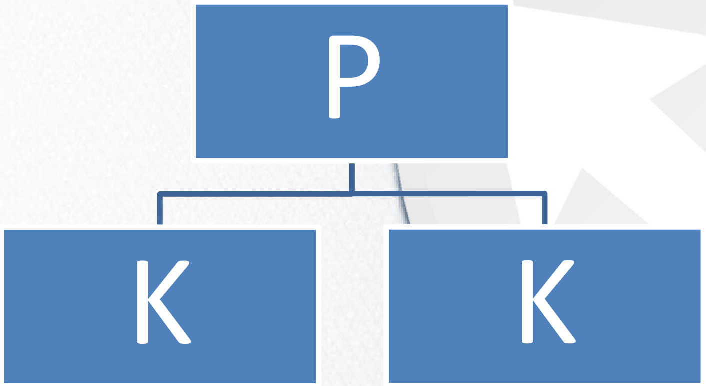
For two class:

$$p_1 + p_2 = 1$$

$$2p_1p_2$$

Minimize! Is the weighted sum Gini index smaller than that of the parent?





Rules

- As derived from trees often have repeated conditions

```
NumCarbon > 3.777 &
SurfaceArea1 > 0.978 &
SurfaceArea1 > 8.404 &
FP009 <= 0.5 &
FP075 <= 0.5 &
NumRotBonds > 1.498 &
NumRotBonds > 1.701
```

Rules and their conditions live on their own, conditions can be adjusted to help bias-variance trade-off

Treatment of Categorical Variables

- Grouped Categories
  - Value treated as related

- Independent Categories
  - Values Treated as Independent

ASSIGNMENT

IMPROVING MODELS

Tree Disadvantages

- List em

Tree Disadvantages

- Model instability (sensitive to data)
  - Derives from each subsequent split is dependent on prior splits
- Less than optimal predictive performance
  - Rectangular regions
- Limited number of outcome values <= number of terminal nodes
- Selection bias toward predictors with higher number of distinct values
- Tuning parameter, C_p
- Splits of correlated variables ambiguous
- Treatment of missing values

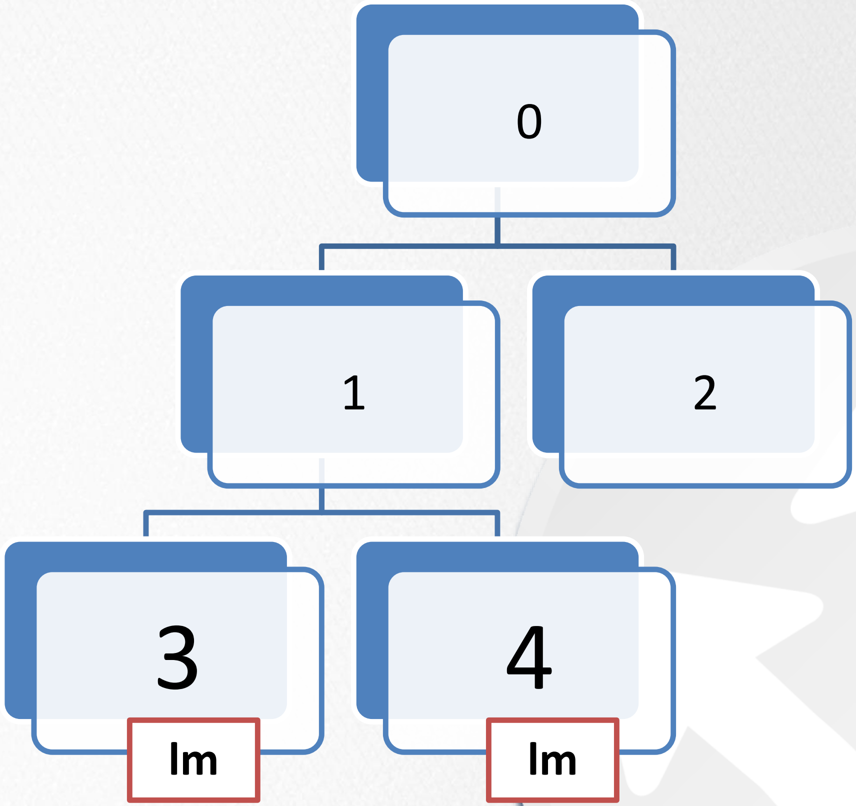
Two Big Ideas

- **Wisdom of the crowds**
  It is better to make estimates from multiple models (***ensembles***) than individual models
  - Better predictions
  - Lower variance for the same model

- **Greed is bad. Patience is good.**
  It is better to slowly approach your solution than arrive at an answer directly

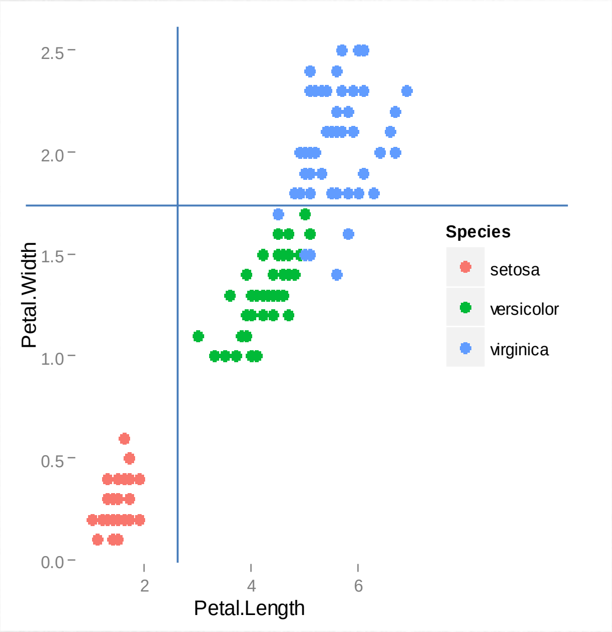
Tree Enhancement

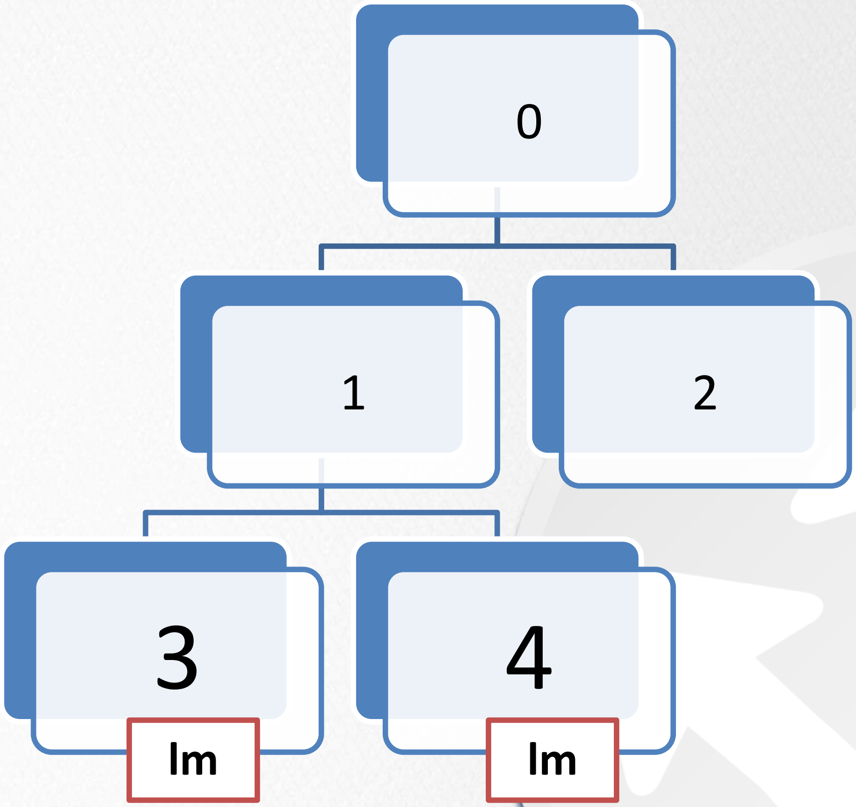
- **Wisdom of the Crowd!**
- Having one value represent the entirety of the node leaves information in the node.
- Function in the node is a simple average
- Use something better
  - **M5** put linear models in nodes of trees



M5 Tree Enhancement (cont.)

- **Greed is bad**
  - linear models are built on the residuals of the tree model.
  - Models are recursive



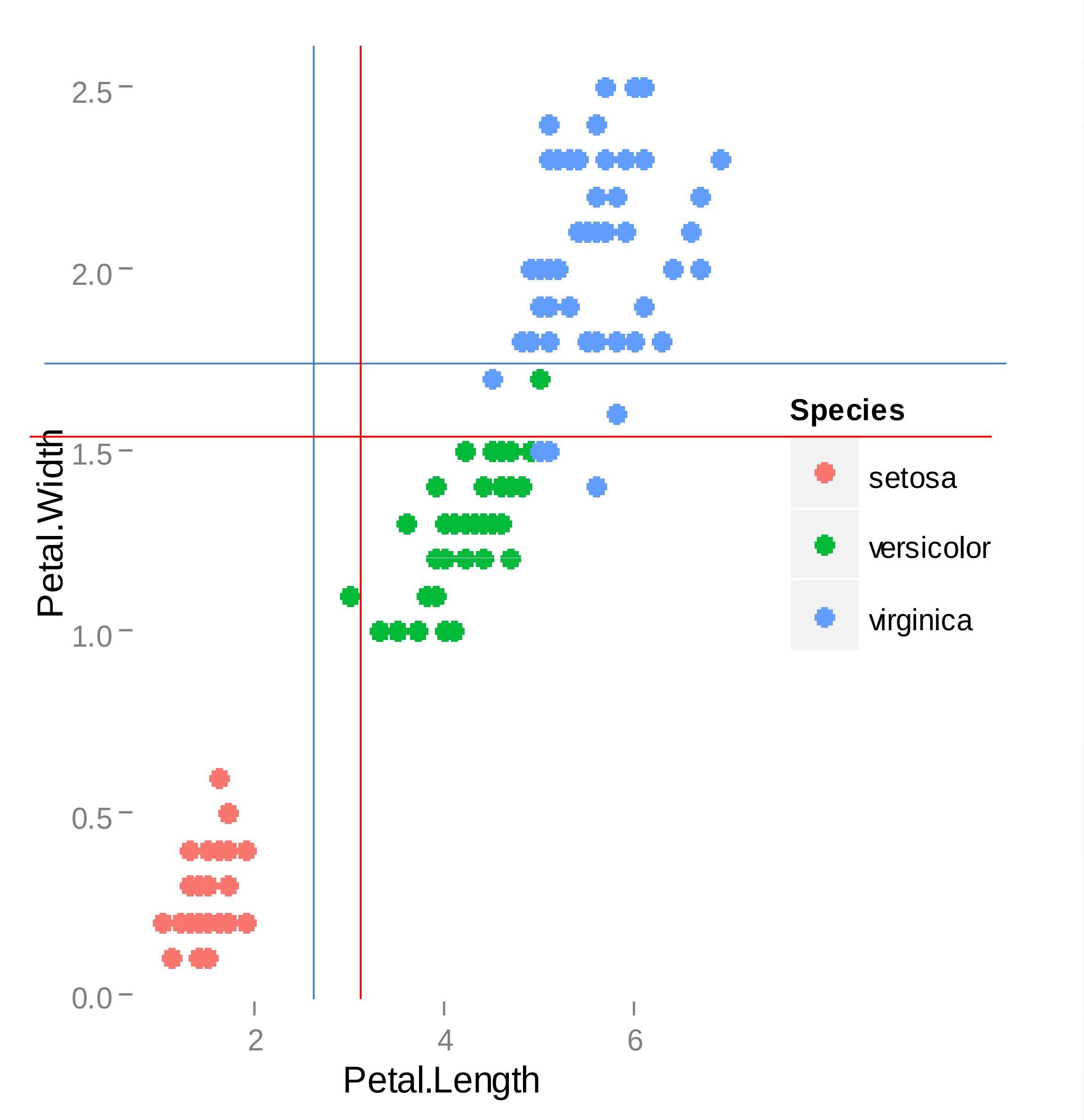


Bagging Models

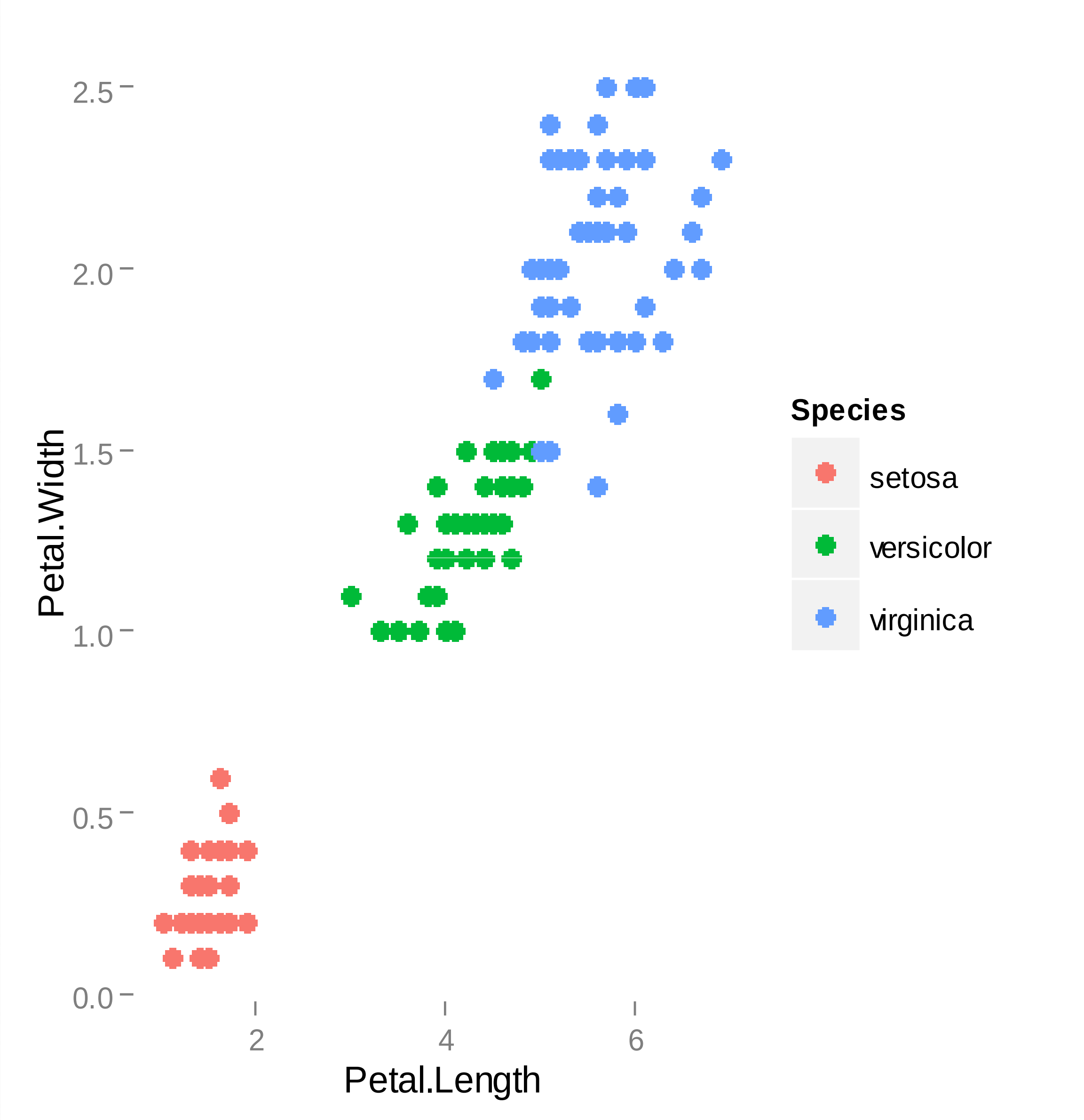
- Brieman:
  "Bagging is a general approach that uses bootstrapping in conjunction with any regression (or classification) model to construct an ensemble."

```
for i = 1 to m do
    Generate a bootstrap sample of the original data
    Train an unpruned tree model on this sample
end
```

$$\hat{y} = \frac{\sum_i \hat{y}_i}{m}$$



2.5
2.0
1.5
1.0
0.5
0.0
Petal.Width
2
4
6
Petal.Length
Species
setosa
versicolor
virginica


2.5
2.0
1.5
1.0
0.5
0.0
2
4
6
Petal.Width
Petal.Length
Species
setosa
versicolor
virginica

Bagging Notes

- Lowers variance
  - Increases stability
  - Has less effect on lower variance models (e.g. linear models)
  - More effect on weak learners

- Disadvantages
  - Computational cost → minor
  - Interpretability

Random Forest

- **Wisdom of the Crowds:** Bagging
- **Greed is bad: consider subset of predictors at each split**

```
Select the number of models to build, m
for i = 1 to m do
    Generate a bootstrap sample of the original data
    Train a tree model on this sample
    for each split do
        Randomly select k (< P) of the original predictors
        Select the best predictor among the k predictors and
        partition the data
    end
    Use typical tree model stopping criteria to determine when a
    tree is complete (but do not prune)
end
```

Tuning Parameter

m_{try} : number of predictors to use at each split

- regression 1/3rd of number predictors
- classification sqrt(number of predictors)

➲ Kuhn: Starting with five values of k that are somewhat evenly spaced across the range from 2 to P.

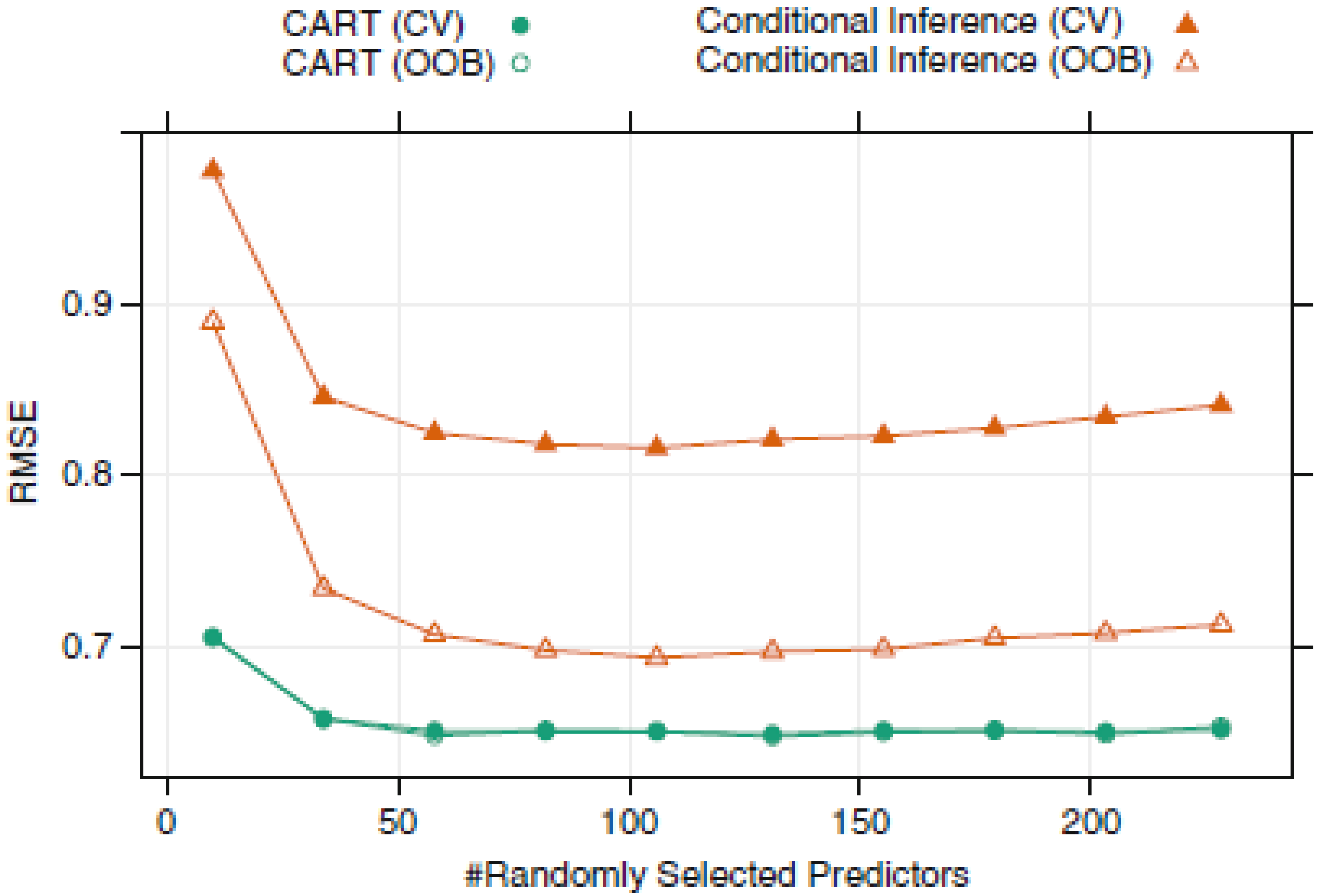


Fig. 8.18: Cross-validated RMSE profile for the CART and conditional inference approaches to random forests

Advantages

- No overfitting
- More trees better (limited by computation time/power only)
- In caret, parameters are considered independently
- Because each learner is selected independently of all previous learners, randomforests is robust to a noisy response
- Computationally efficient -- each tree built on subset of predictors at each split.
- Use any tree variants as "base learner": CART, ctree, etc

APPENDIX

EXAMPLE OF ML ALGORITHM(S)

- Spam Filter
- handwriting recognition (svm)
- Traffic engineering (lights)
- Weather prediction
- Sentiment analysis (social media)
- Netflix Recommender
- Fraud detection (Visa)
- Imaging processing
- (network) Intrution detection
- Self-driving cars