Regression Trees and Random Forests

Lecture 5 – Mark Segal – Epidemiology and Biostatistics

Tree-structured Methods

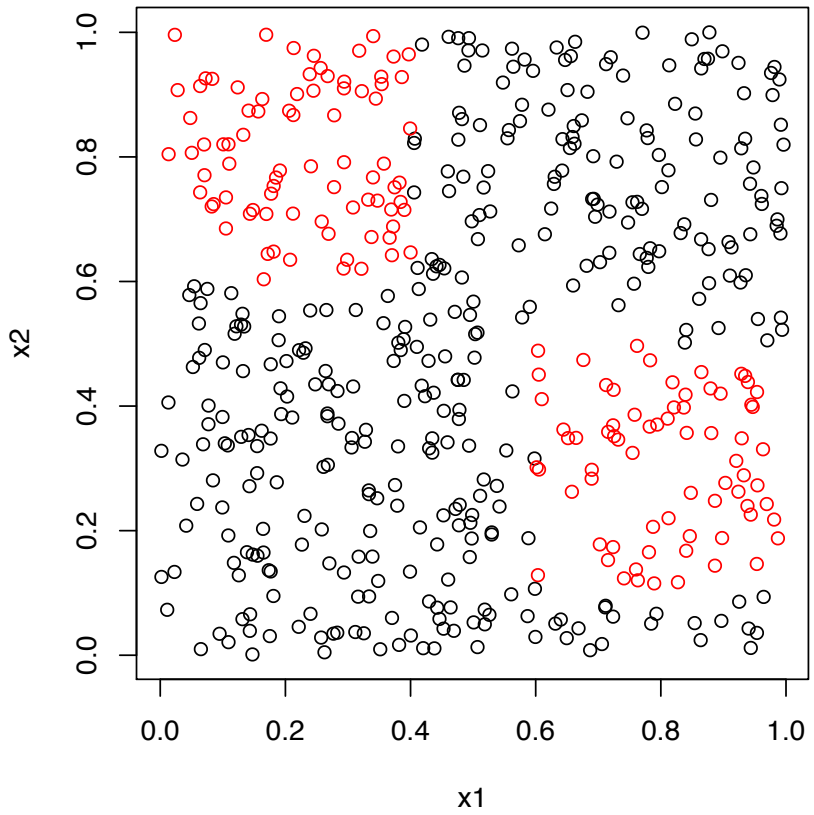
- Breiman, Friedman, Olshen, Stone (1984) CART
- Popularized tree-structured techniques
- Handles classification ((multi) categoric outcomes) and regression (continuous outcomes) problems
  - subsequent extensions to survival, multiple outcomes
- Trees grown by recursively partitioning feature space using univariate binary (into two groups) splitting
  - allowable splits determined by covariate type
  - split criterion function determined by outcome type
- Tree size determined by cost-complexity pruning
  - captures important interactions; avoids overfitting

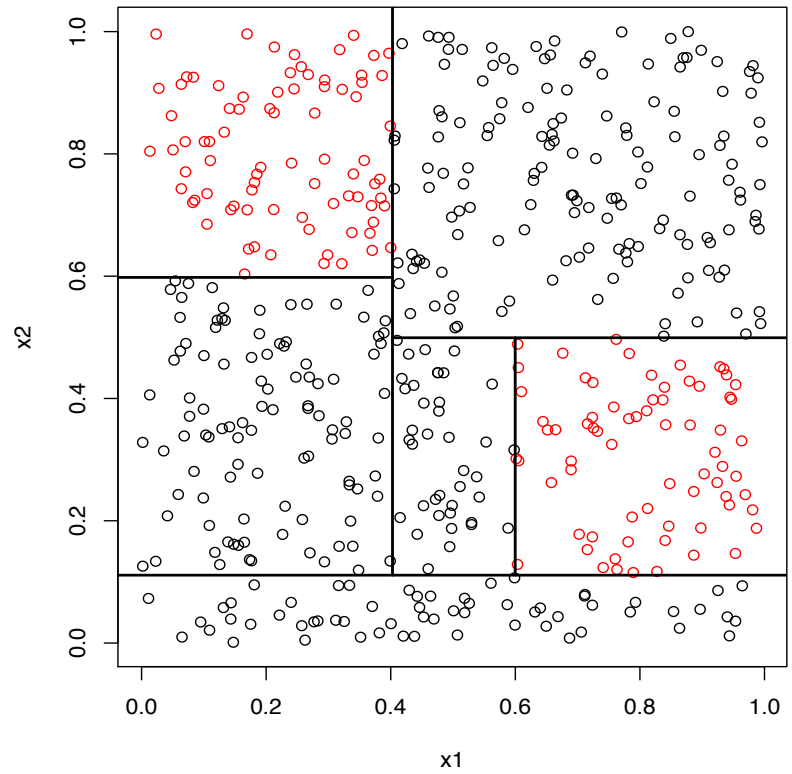
CART Paradigm

Tree construction involves four constituent components:

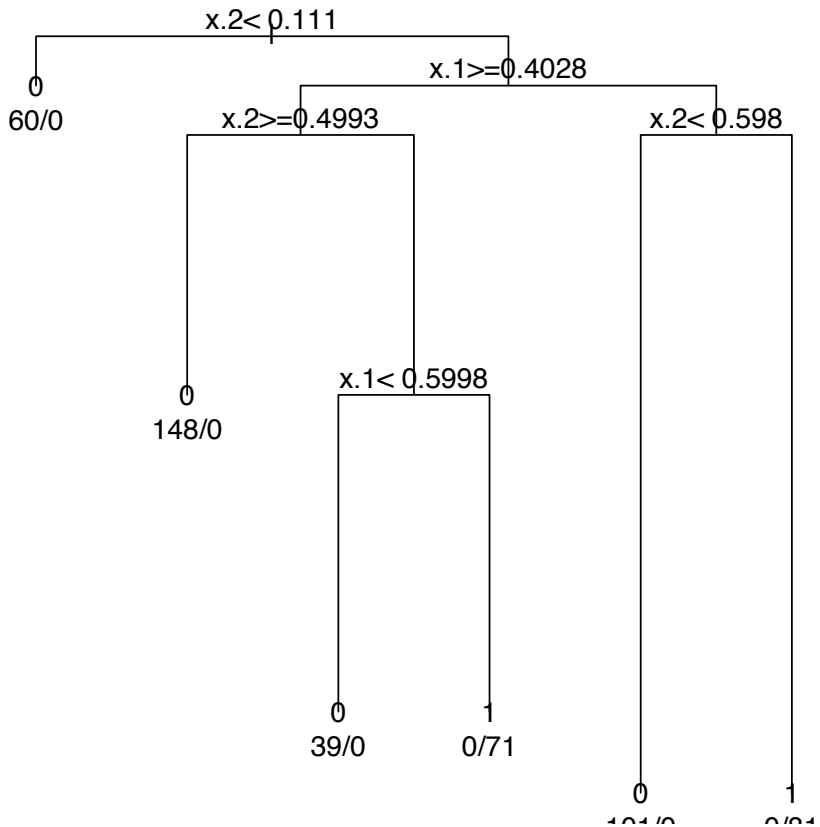
1. A set of yes/no questions, splits, framed in terms of covariates that partition the predictor space. Recursively applying these splits yields a (binary) tree. Subsamples created by assigning cases according to splits are termed nodes.
2. A split function that can be evaluate for any split of any node used to assess the worth of the competing splits.
3. A means for determining appropriate tree size.
4. Statistical summaries for the nodes of the tree.

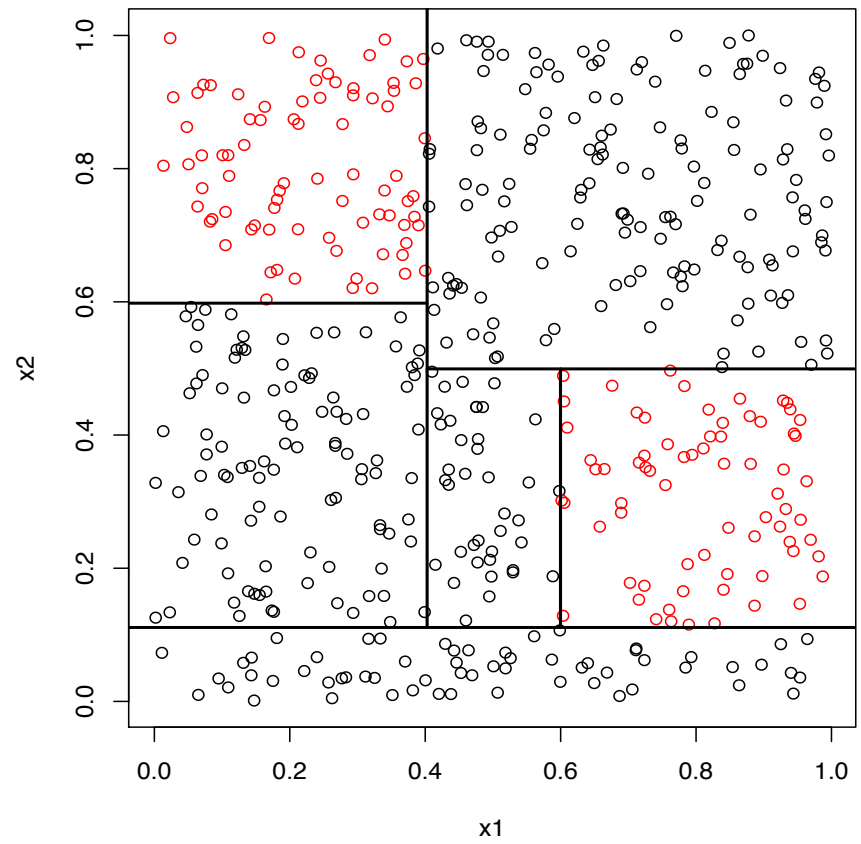
- For continuous covariates all order preserving splits are allowed: $X_j \leq c$ and $X_j > c$ How many such splits?



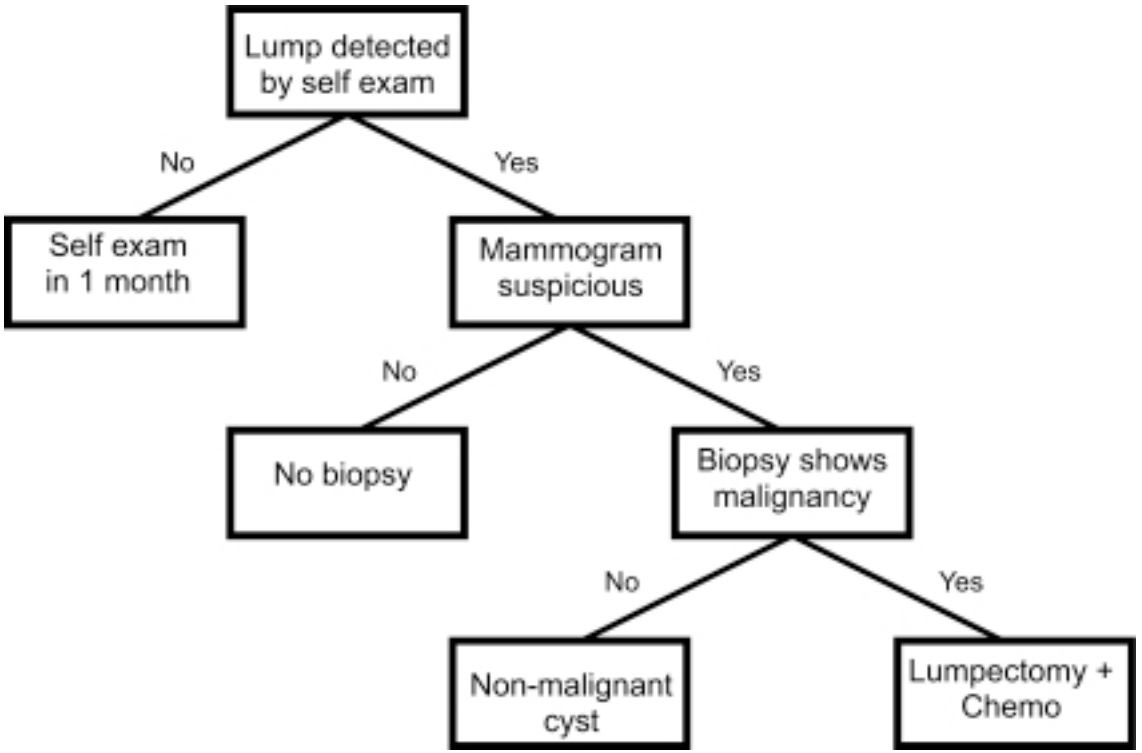


- Feature space recursive partitioning can be depicted via a tree schematic:





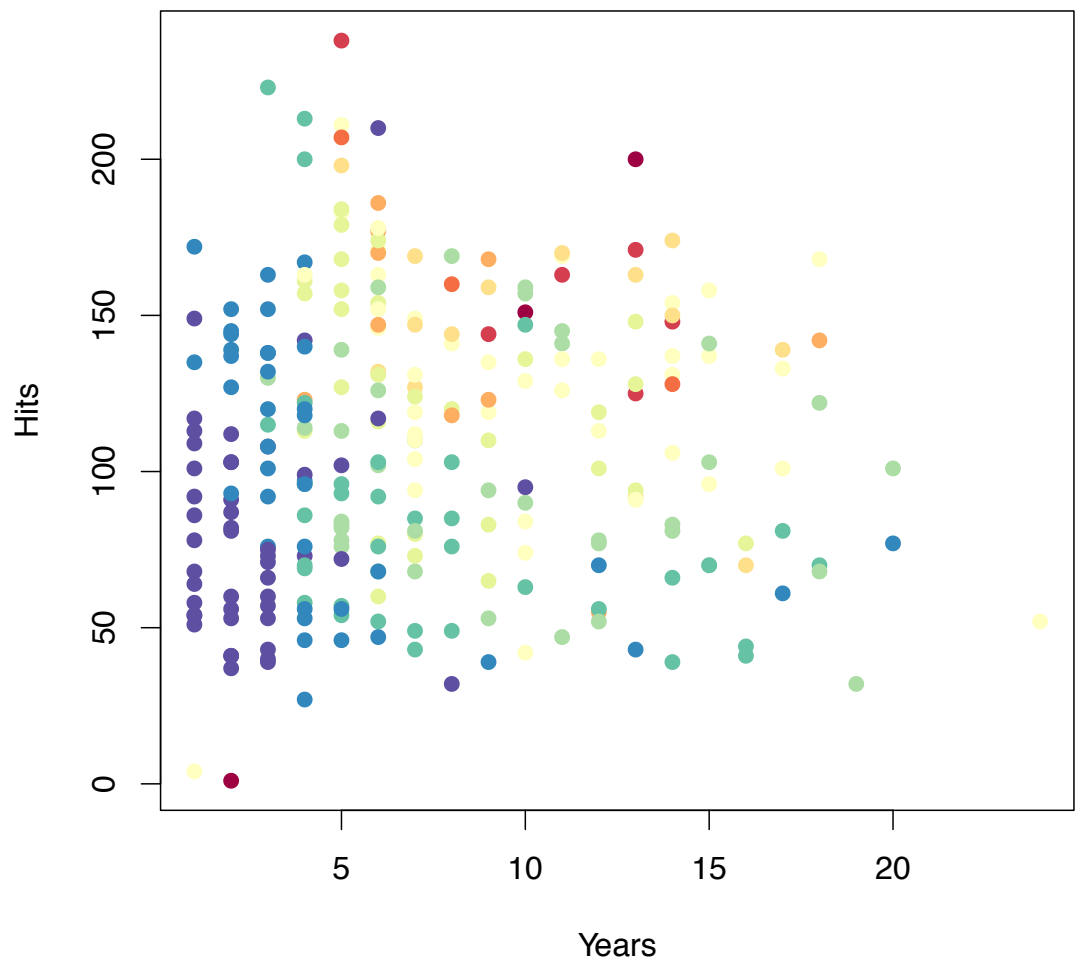
- Interpretability derives from mimicking decision process



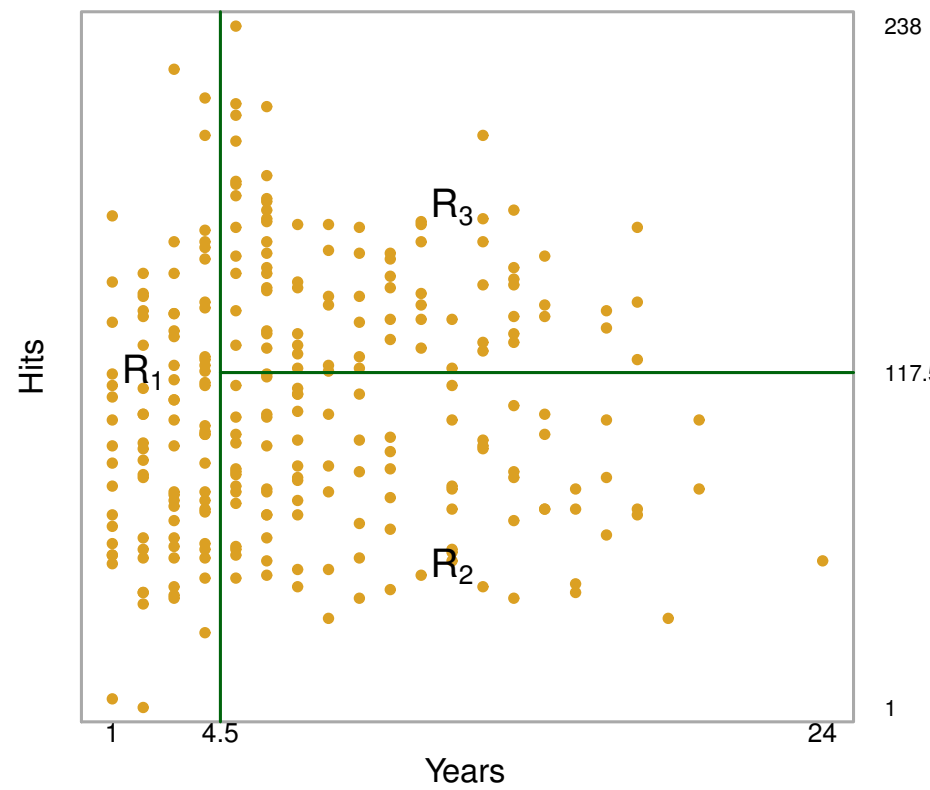
- For categorical covariates all binary splits into disjoint category subsets are allowed. For such a covariate with k levels how many such splits?

Baseball salary data: how would you stratify it?

Salary is color-coded from low (blue, green) to high (yellow,red)



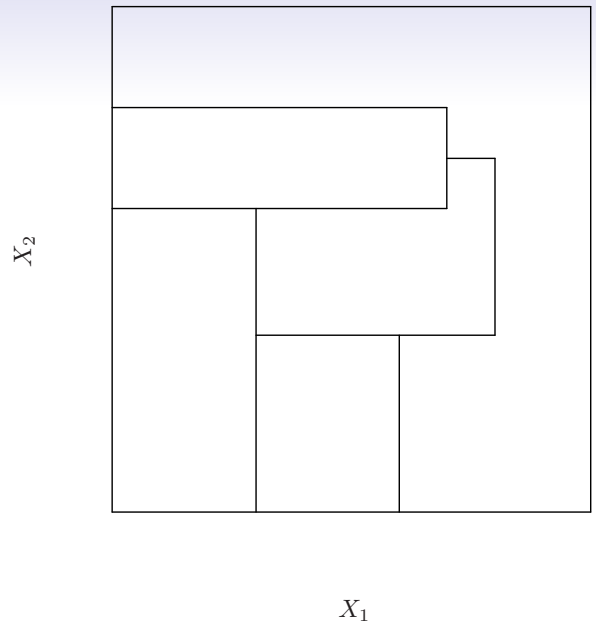
Decision tree for these data

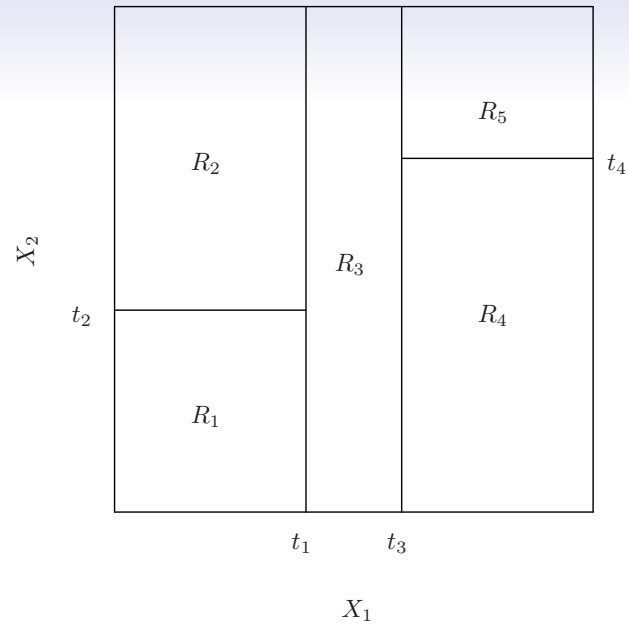


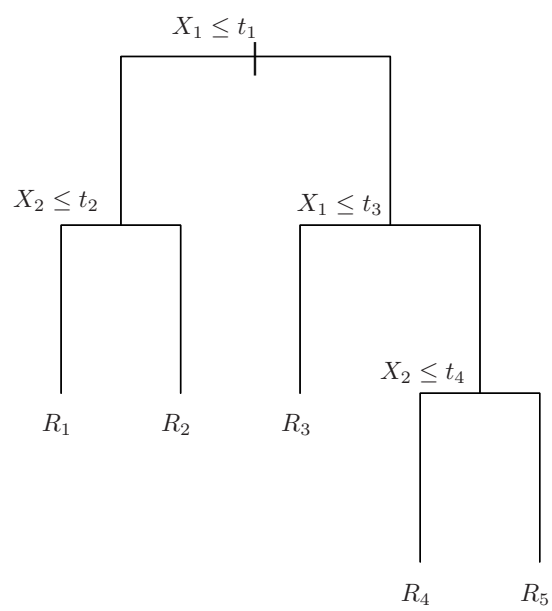
- At a given internal node, the label (of the form $X_j < t_k$) indicates the left-hand branch emanating from that split, and the right-hand branch corresponds to $X_j \geq t_k$. For instance, the split at the top of the tree results in two large branches. The left-hand branch corresponds to `Years`<4.5, and the right-hand branch corresponds to `Years`>=4.5.

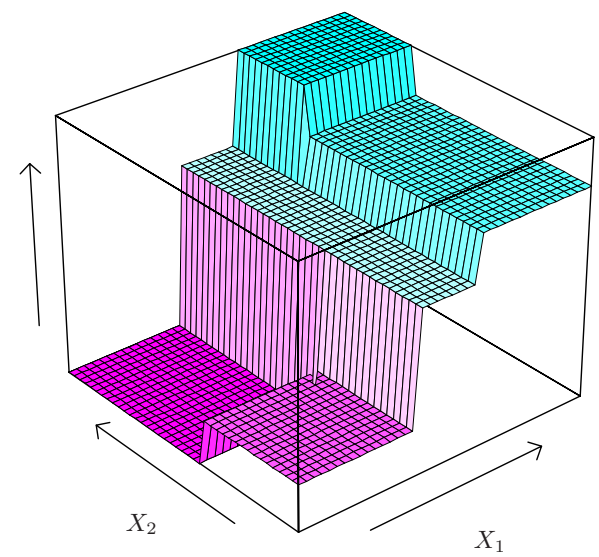
- The tree has two internal nodes and three terminal nodes, or leaves. The number in each leaf is the mean of the response for the observations that fall there.

- Overall, the tree stratifies or segments the players into three regions of predictor space: $R_1 = \{\text{X} \mid$ `Years`$< 4.5\}$, $R_2 = \{\text{X} \mid$ `Years`$>=4.5$, `Hits`$<117.5\}$, and $R_3 = \{\text{X} \mid$ `Years`$>=4.5$, `Hits`$>=117.5\}$.


X_2
X_1


R_2
R_5
t_4
R_3
X_2
t_2
R_4
R_1
t_1
t_3
X_1


$X_1 \leq t_1$
$X_2 \leq t_2$
$X_1 \leq t_3$
$X_2 \leq t_4$
R_1
R_2
R_3
R_4
R_5


X_2
X_1

Split Criterion Functions

- Classification
  - Gini diversity index; cross-entropy (deviance)
    - not misclassification error
- Regression
  - squared error deviations (*cf* t-statistic)
  - least absolute deviations
- Survival
  - log-rank statistic

Sum-of-squares Split Function

Node $g = \{(\mathbf{x}'_i, y_i)\}$, $n_g = |g|$, $\bar{y}(g) = \frac{1}{n_g} \sum_{i \in g} y_i$

$SS(g) = \sum_{i \in g} (y_i - \bar{y}(g))^2$

Suppose split s partitions g into nodes g_L and g_R

SS split fn: $\phi(s, g) = SS(g) - SS(g_L) - SS(g_R)$

Best split s^* of g: $\phi(s^*, g) = \max_{s \in \Omega} \phi(s, g)$

where Ω is the set of all allowable splits s of g.

Since $\phi \geq 0$ recursive splitting creates smaller nodes of progressively increasing homogeneity.

Determining Tree Size

$R(g)$ *cost* of node g; for example $R(g) = SS(g)$
$R(G)$ cost of tree G: $R(G) = \sum_{g \in \widetilde{G}} R(g)$
where $\widetilde{G}$ is the set of *terminal* nodes
Complexity of G: $|\widetilde{G}|$, number of terminal nodes
Cost-complexity: $R_\alpha(G) = R(G) + \alpha\,|\widetilde{G}|;\ \alpha \geq 0$

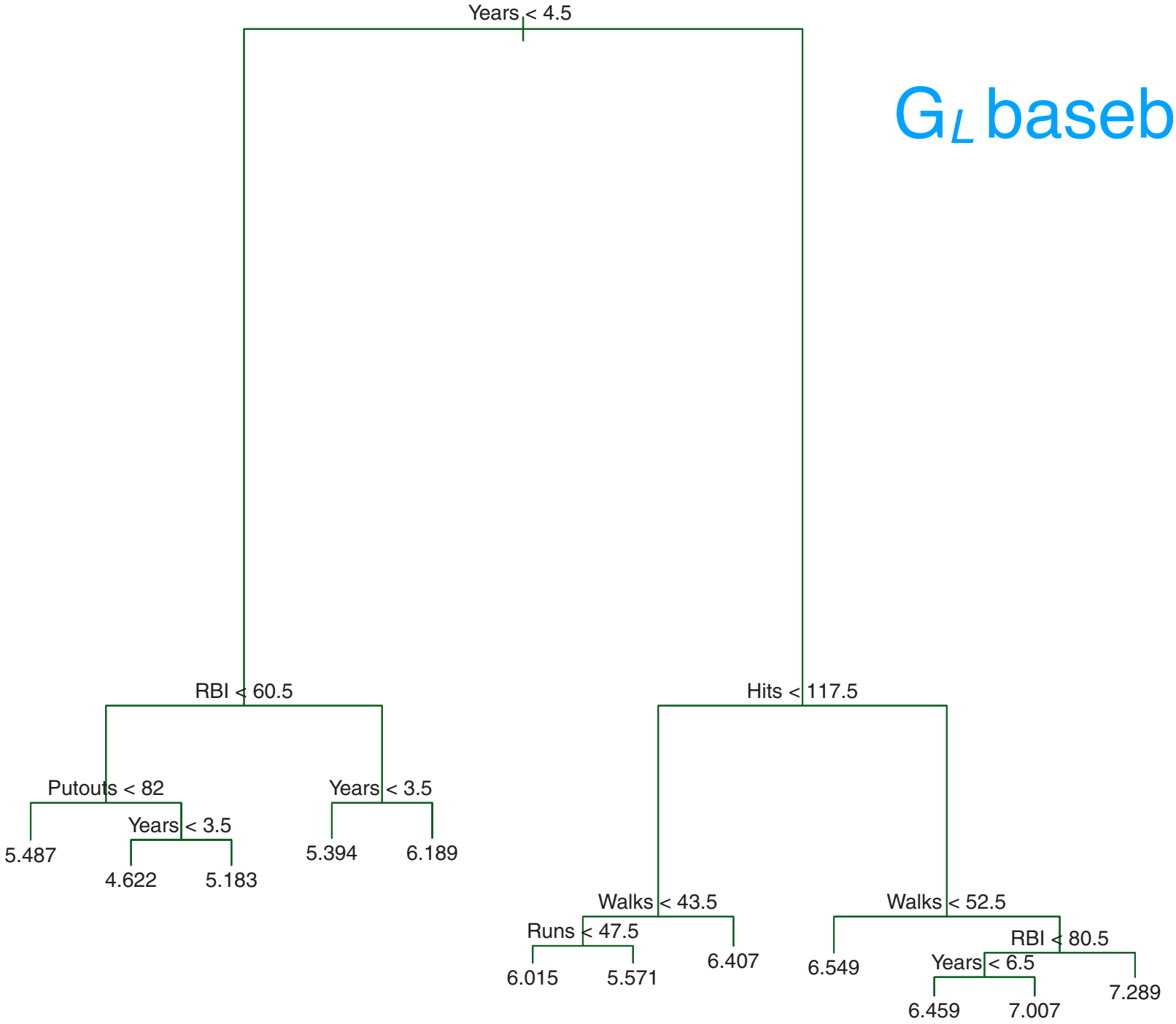
Simultaneously minimize cost *and* complexity:
large trees have small cost, high complexity $\Rightarrow$
solely minimizing cost leads to overfitting.

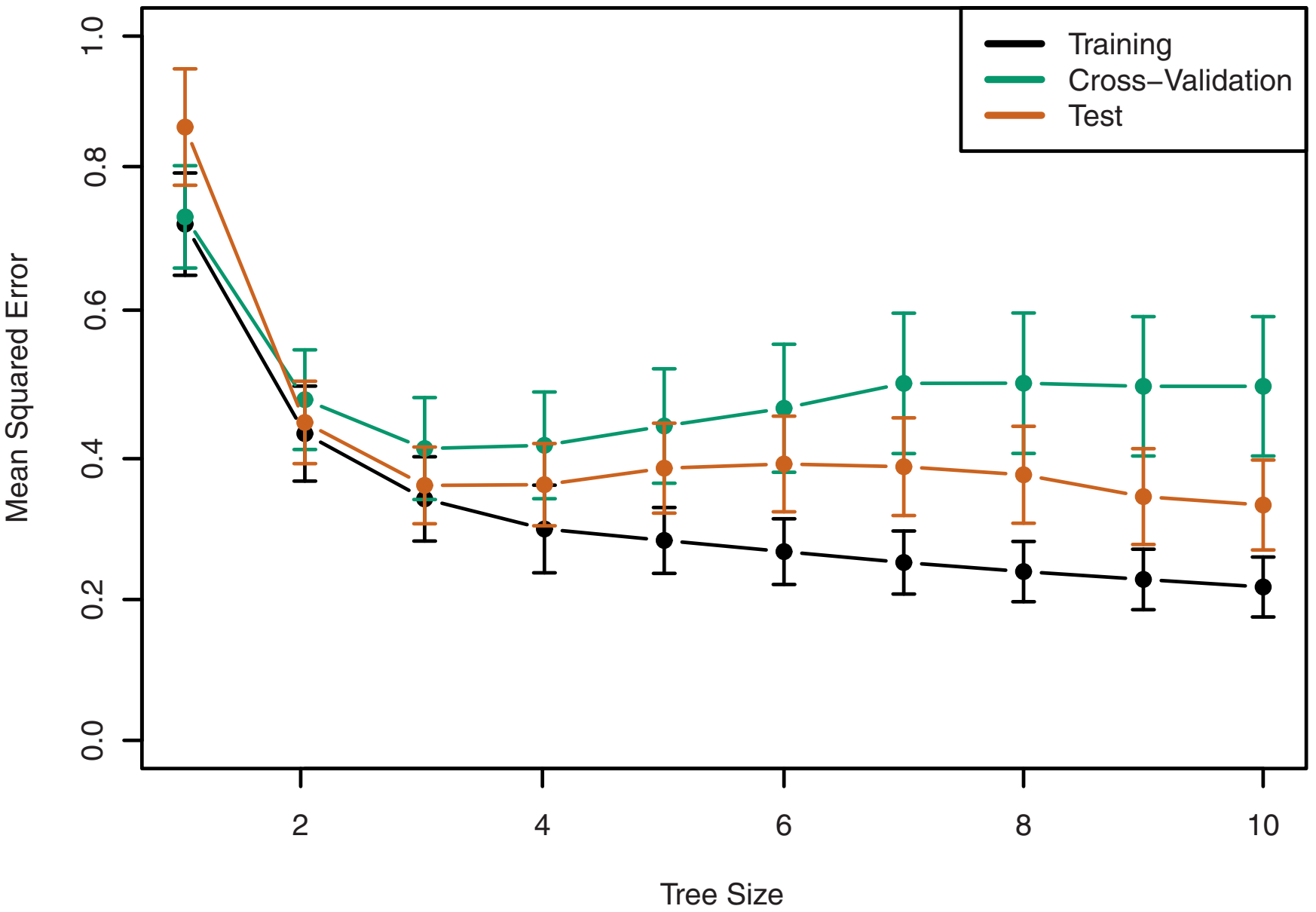
Grow a large initial tree G_L.
Find subtree G_α of G_L that minimizes $R_\alpha(G)$.
If α is small, the penalty for large $|\widetilde{G}|$ is small
and hence G_α will be large.
As $\alpha \nearrow$, $|\widetilde{G}_\alpha| \searrow$.
For α sufficiently large, $|\widetilde{G}_\alpha| = 1$ and the
minimal cost-complexity tree is the *root* node
$\Rightarrow$ nested sequence of optimally pruned subtrees.

Pick tree minimizing $R_\alpha(G)$ via cross-validation.

Size determination is a critical consideration.

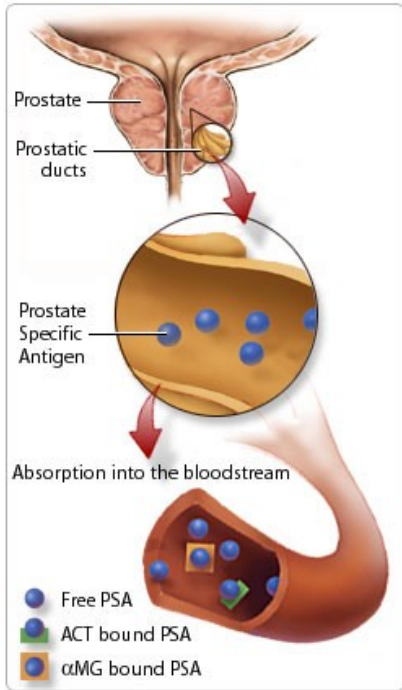
G_L baseball data




Training
Cross–Validation
Test
Mean Squared Error
0.0
0.2
0.4
0.6
0.8
1.0
2
4
6
8
10
Tree Size

Prostate cancer example:

E.g., suppose that we are studying the level of prostate-specific antigen (PSA), which is often elevated in men who have prostate cancer. We look at $n = 97$ men with prostate cancer, and $p = 8$ clinical measurements.[1] We are interested in identifying a small number of predictors, say 2 or 3, that drive PSA

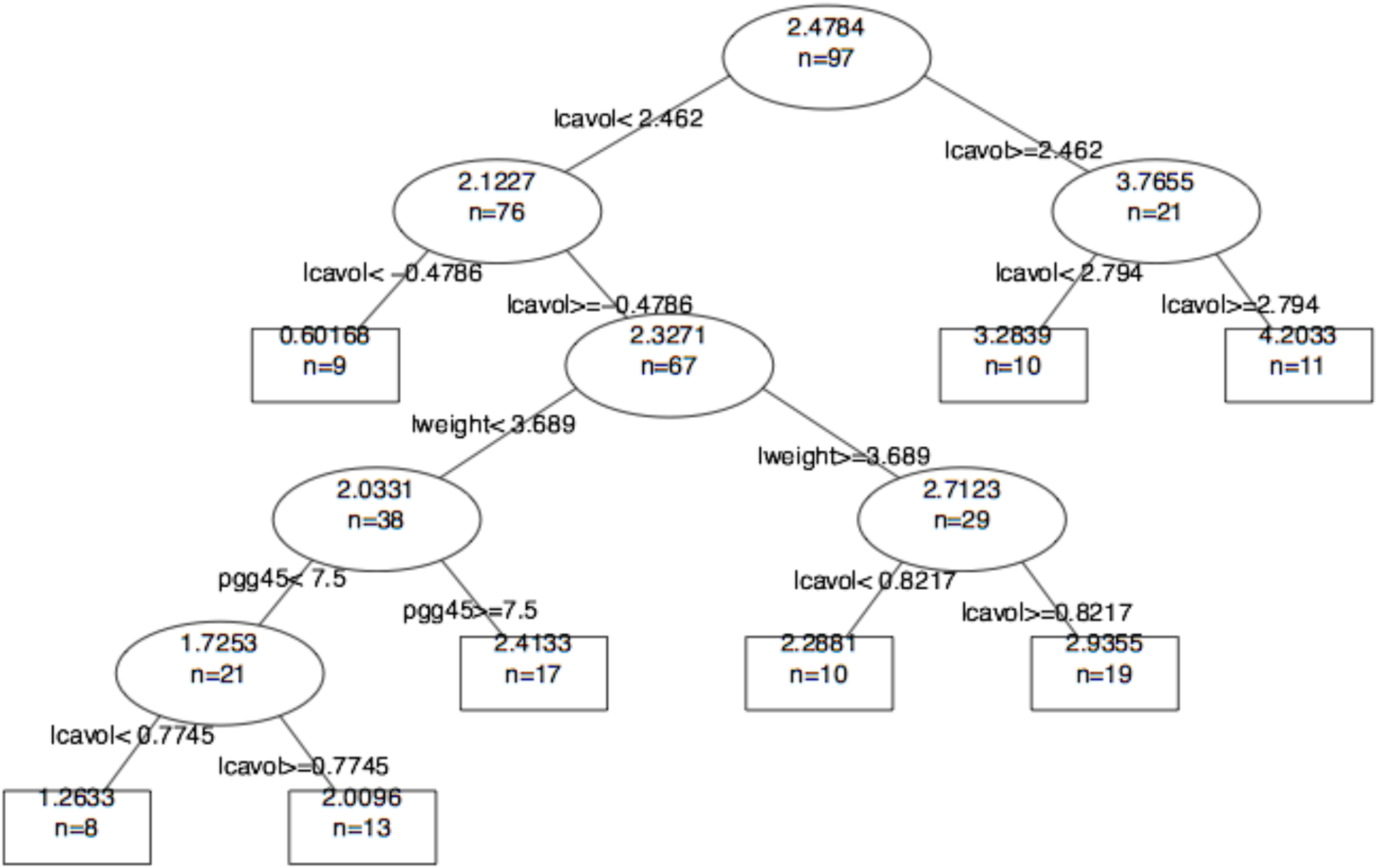


[2]

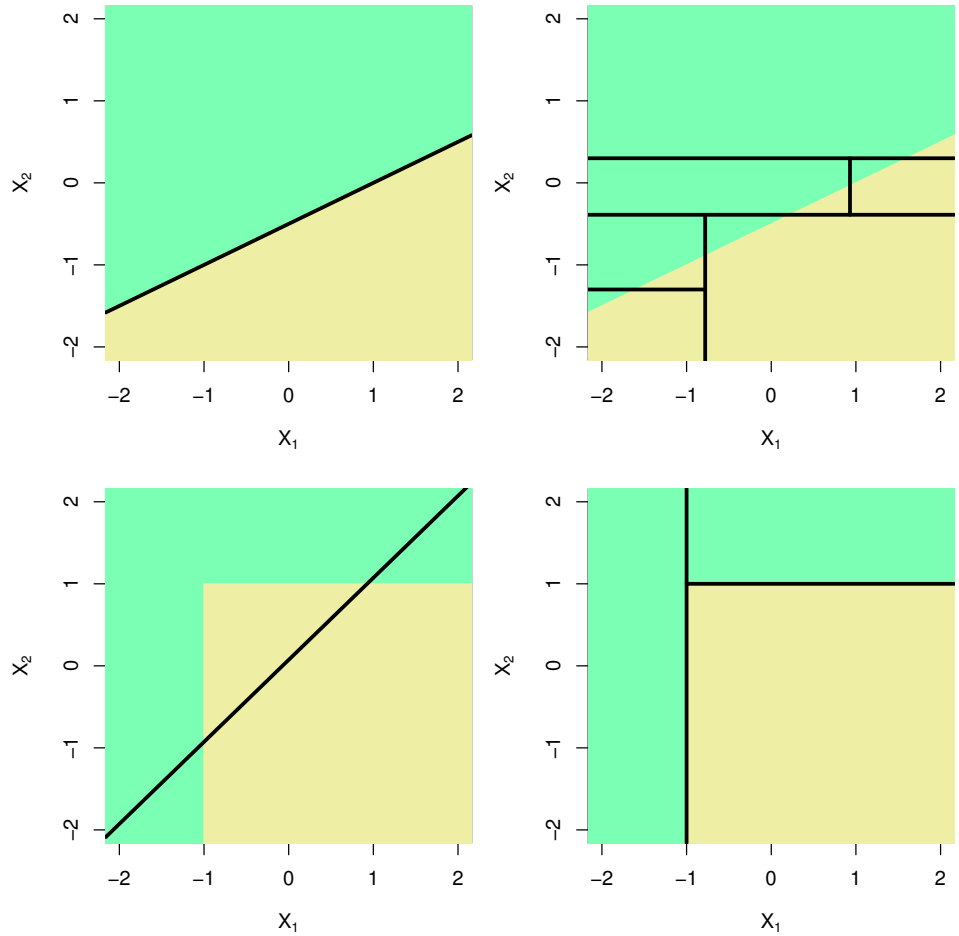
[1] Data from Stamey et al. (1989), "Prostate specific antigen in the diag..."

[2] Figure from `http://www.mens-hormonal-health.com/psa-score.html`

Outcome: log(PSA)



Trees Versus Linear Models



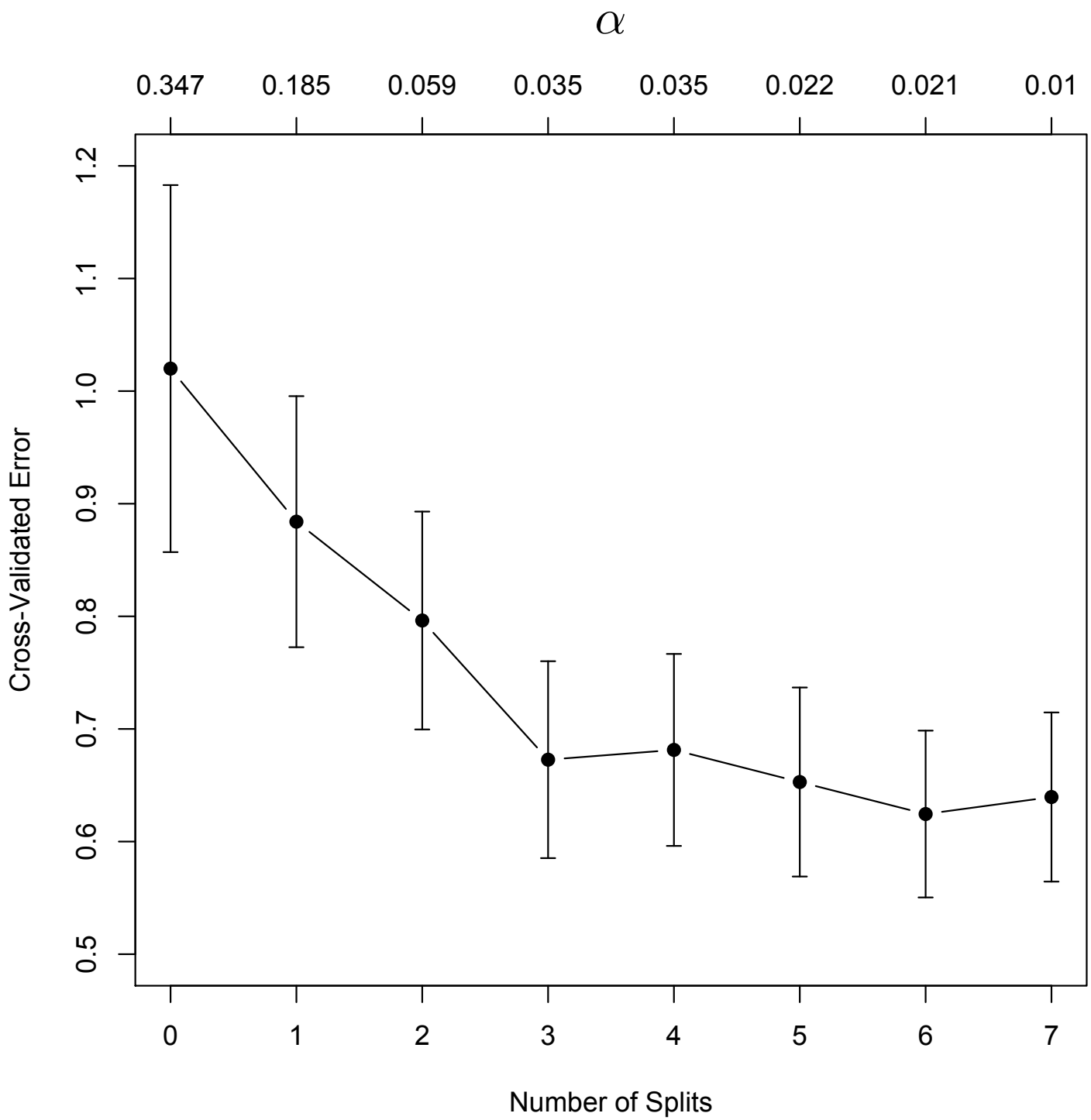
Top Row: True linear boundary; Bottom row: true non-linear boundary.

Left column: linear model; Right column: tree-based model

```
Node number 1: 97 observations,    complexity param=0.347
  mean=2.48, MSE=1.32
  left son=2 (76 obs) right son=3 (21 obs)
  Primary splits:
      lcavol  < 2.46   to the left,  improve=0.347, (0 missing)
      svi     < 0.5    to the left,  improve=0.321, (0 missing)
      lcp     < 0.136  to the left,  improve=0.277, (0 missing)
      pgg45   < 8      to the left,  improve=0.252, (0 missing)
      gleason < 6.5    to the left,  improve=0.234, (0 missing)
  Surrogate splits:
      lcp     < 1.5    to the left,  agree=0.887, adj=0.476, (0 split)
      svi     < 0.5    to the left,  agree=0.856, adj=0.333, (0 split)
      age     < 76.5   to the left,  agree=0.804, adj=0.095, (0 split)
      pgg45   < 92.5   to the left,  agree=0.804, adj=0.095, (0 split)
      gleason < 8.5    to the left,  agree=0.794, adj=0.048, (0 split)
```

Surrogate splits used for handling missing data and determining variable importances:

```
    lcavol      lcp      svi    pgg45      age lweight gleason     lbph
        47       15       10        6        6        6        5        5
```

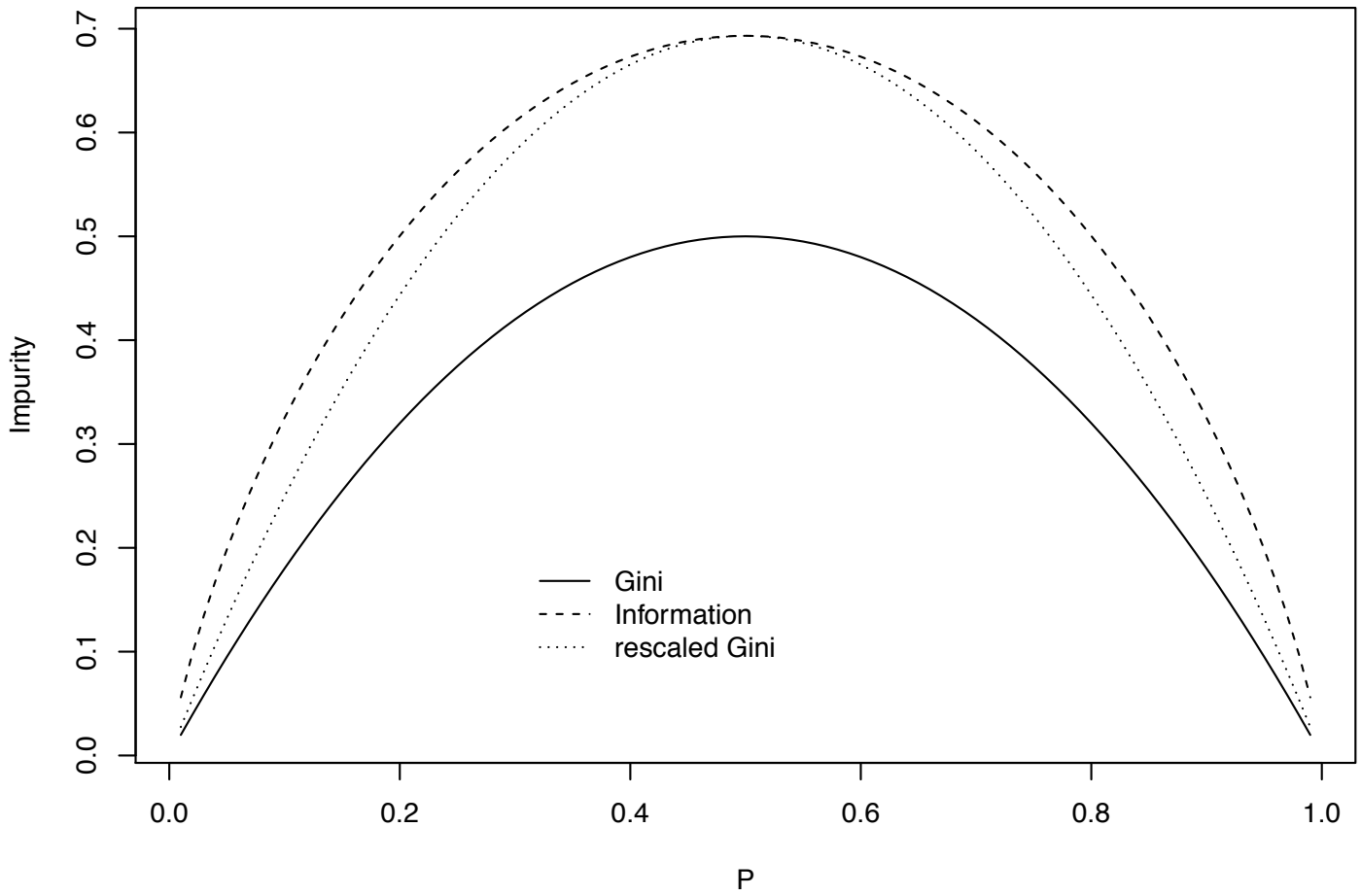

α
0.347
0.185
0.059
0.035
0.035
0.022
0.021
0.01
Cross-Validated Error
0.5
0.6
0.7
0.8
0.9
1.0
1.1
1.2
0
1
2
3
4
5
6
7
Number of Splits

Classification Trees

- Pertain to categorical outcomes
- Utilize same growing (binary splitting) and pruning strategies as described for regression trees
- Modify loss function for evaluating competing splits:
  - Gini diversity index $$G = \sum_{k=1}^{K} \hat{p}_{mk}(1 - \hat{p}_{mk})$$

$\hat{p}_{mk}$: proportion of observations in class *k* and node *m*

  - entropy (information) $$D = -\sum_{k=1}^{K} \hat{p}_{mk} \log \hat{p}_{mk}$$
  - not misclassification error $$E = 1 - \max_{k}(\hat{p}_{mk})$$
    - but used for pruning



Issue with E is that risk reduction ϕ is linear:
does not differentiate between splits that produce
(i) purities of 85% and 50% vs (ii) 70% and 70%
Which is preferable?

Classification Tree Features

- Variable misclassification costs
- Prescribed class priors / probabilities
- In fact, the manner in which variable misclassification costs are incorporated into the loss function is via altering class priors
  - see the rpart() vignette for details

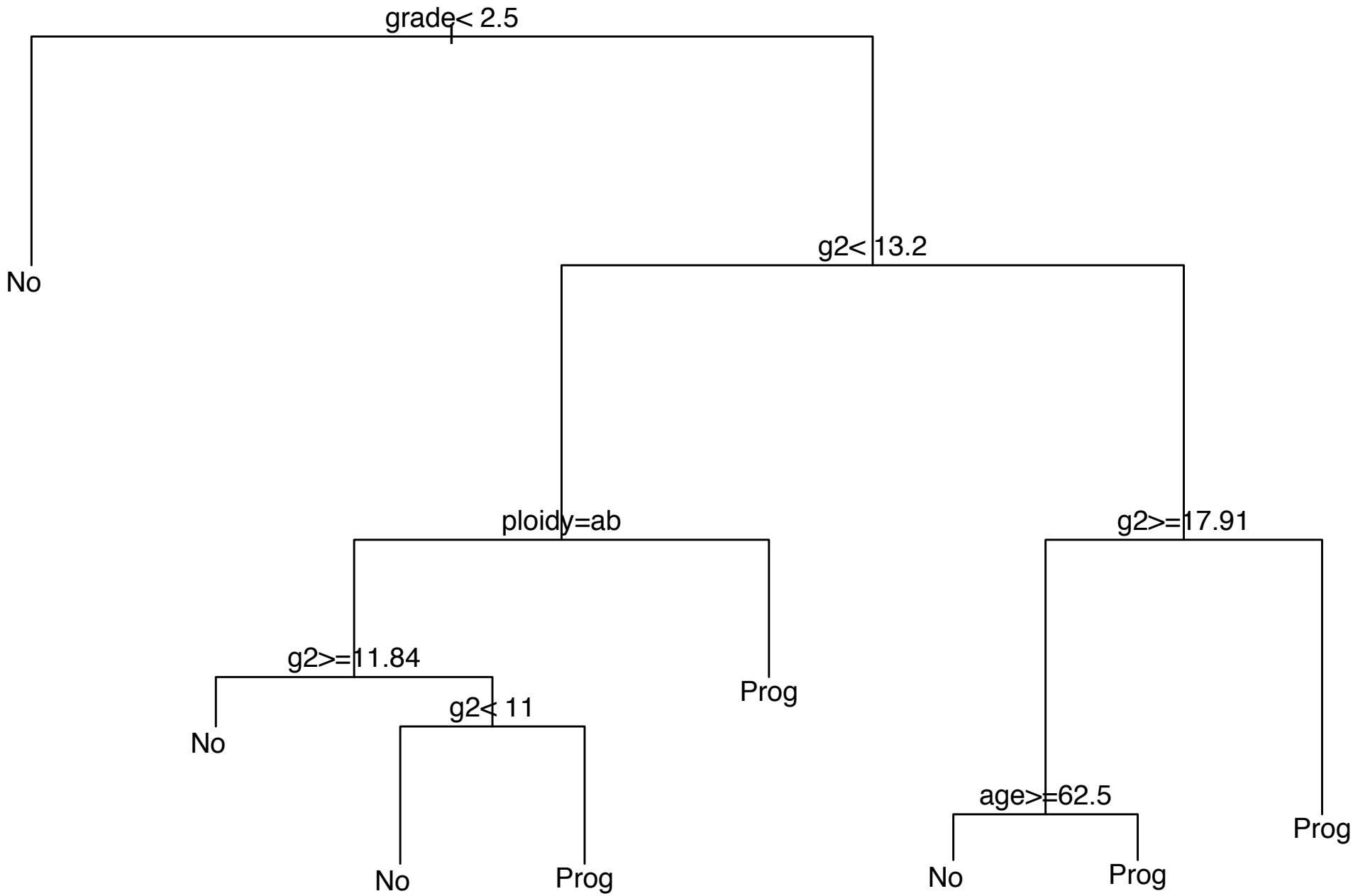
Classification Tree Example

Stage C prostate cancer
Class outcome: progression status

pgtime	time to progression, or last follow-up free of progression
pgstat	status at last follow-up (1=progressed, 0=censored)
age	age at diagnosis
eet	early endocrine therapy (1=no, 0=yes)
ploidy	diploid/tetraploid/aneuploid DNA pattern
g2	% of cells in G_2 phase
grade	tumor grade (1-4)
gleason	Gleason grade (3-10)

Time-to-event outcomes: survival trees

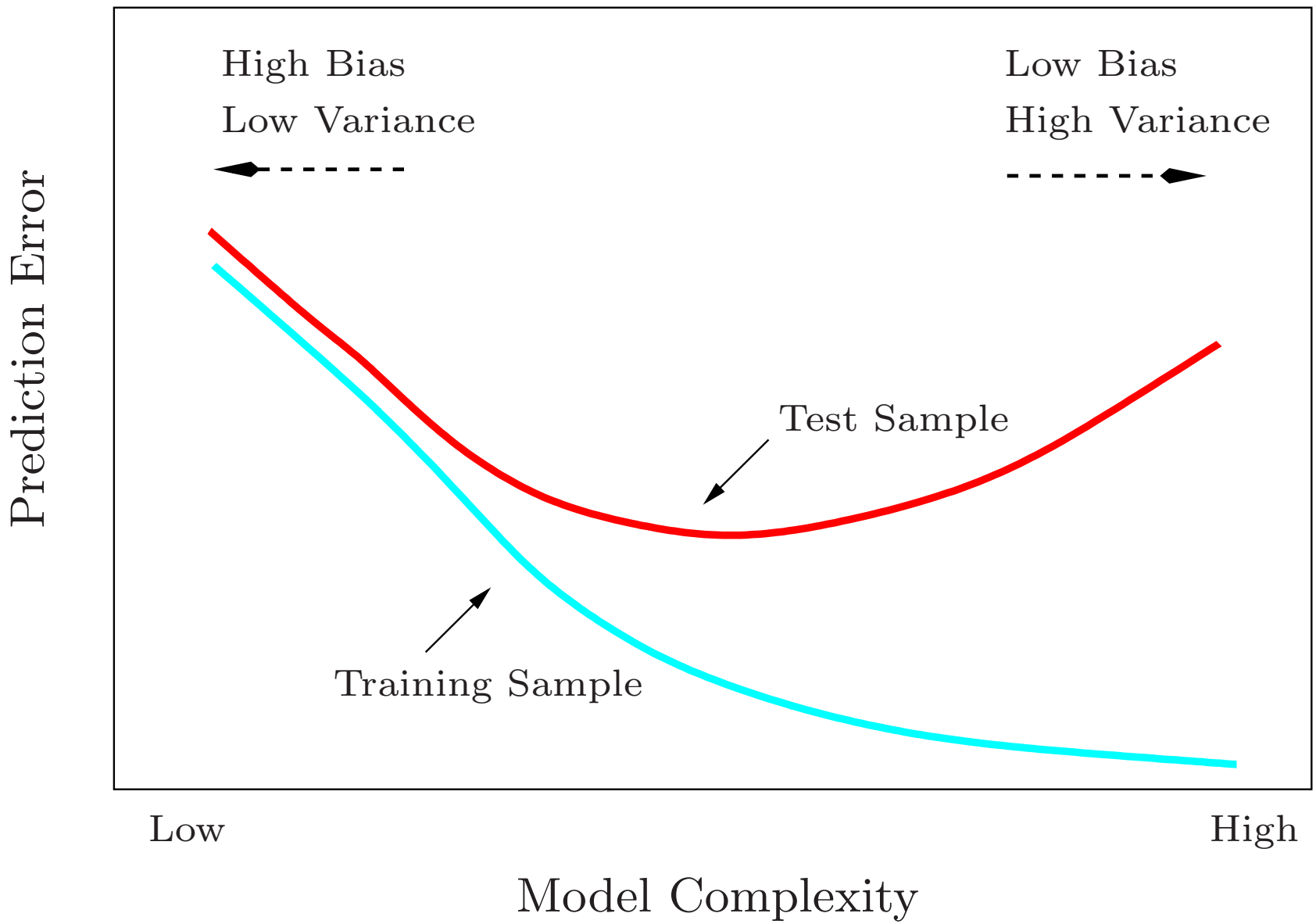
```
> progstat <- factor(stagec$pgstat, levels = 0:1, labels = c("No", "Prog"))
> cfit  <- rpart(progstat ~  age + eet + g2 + grade + gleason + ploidy,
                 data = stagec, method = 'class')
> print(cfit)
n= 146

node), split, n, loss, yval, (yprob)
      * denotes terminal node

 1) root 146 54 No (0.6301370 0.3698630)
   2) grade< 2.5 61  9 No (0.8524590 0.1475410) *
   3) grade>=2.5 85 40 Prog (0.4705882 0.5294118)
     6) g2< 13.2 40 17 No (0.5750000 0.4250000)
      12) ploidy=diploid,tetraploid 31 11 No (0.6451613 0.3548387)
        24) g2>=11.845 7  1 No (0.8571429 0.1428571) *
        25) g2< 11.845 24 10 No (0.5833333 0.4166667)
          50) g2< 11.005 17  5 No (0.7058824 0.2941176) *
          51) g2>=11.005 7  2 Prog (0.2857143 0.7142857) *
      13) ploidy=aneuploid 9  3 Prog (0.3333333 0.6666667) *
     7) g2>=13.2 45 17 Prog (0.3777778 0.6222222)
      14) g2>=17.91 22  8 No (0.6363636 0.3636364)
        28) age>=62.5 15  4 No (0.7333333 0.2666667) *
        29) age< 62.5 7  3 Prog (0.4285714 0.5714286) *
      15) g2< 17.91 23  3 Prog (0.1304348 0.8695652) *
```


grade< 2.5
No
g2< 13.2
ploidy=ab
g2>=17.91
g2>=11.84
Prog
g2< 11
No
age>=62.5
Prog
No
Prog
No
Prog

Tree Attributes

- Interpretability; graphical display; mimic decision process
- Invariance under monotone transformation
  - finesses normalization complexities
- Handling of mixed covariate types
- Handling of missing data
- Provision of covariate importance scores
  - Filtering, variable selection
- Devoid of parametric assumptions
- Insensitivity to outliers
- Predictive performance ?

Predictive Performance



Breiman Mantra

- Better the model fits, the more sound the inference
- Conventional models and CART tend to fit poorly
- Fit measured by prediction error (*PE*)
- Substantial gains in *PE* can be achieved by using ensembles of (weak, unstable) predictors
  - in particular, individual trees

Bagging

- Bootstrap aggregation - bagging - general purpose procedure for reducing variance of prediction method
- Based on the simple fact that the variance of the mean of *B iid* observations is reduced (by a factor of *B*)
- Bootstrap used to generate *id* datasets
  - prediction method trained on each and averaged:

$$\hat{f}_{\text{bag}}(x) = \frac{1}{B}\sum_{b=1}^{B} \hat{f}^{*b}(x)$$

Random Forests

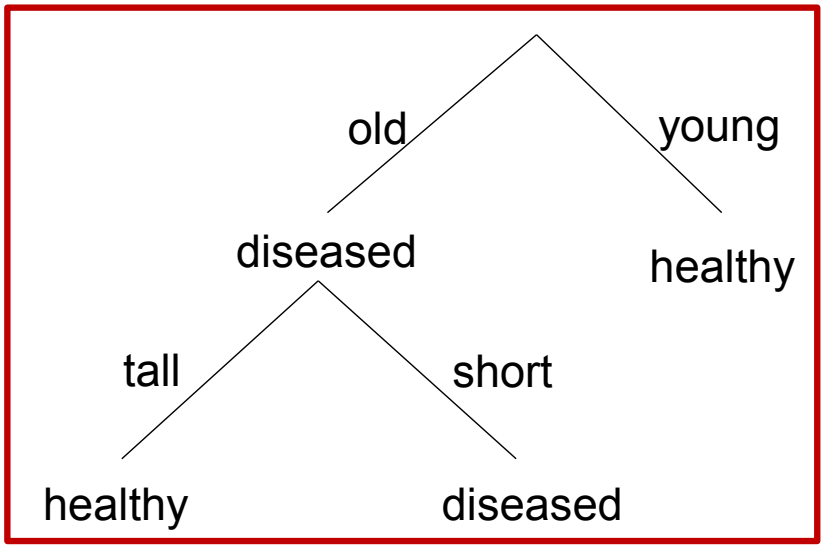
- Breiman (2001a,b)
- Have become a forefront prediction technique
- Gains in prediction performance over individual trees
  - *PE variance* reduced by averaging over the *randomness-injected* bootstrap ensemble of trees
    - Individual trees grown to large / maximal depth
      - Major departure from CART paradigm
  - *Seemingly*, averaging over the ensemble *more than* compensates for increased individual tree variability
- Bootstrapping provides built-in estimate of *PE: OOB*
- Interpretation aided by variable importance measures

Random Forest Construction

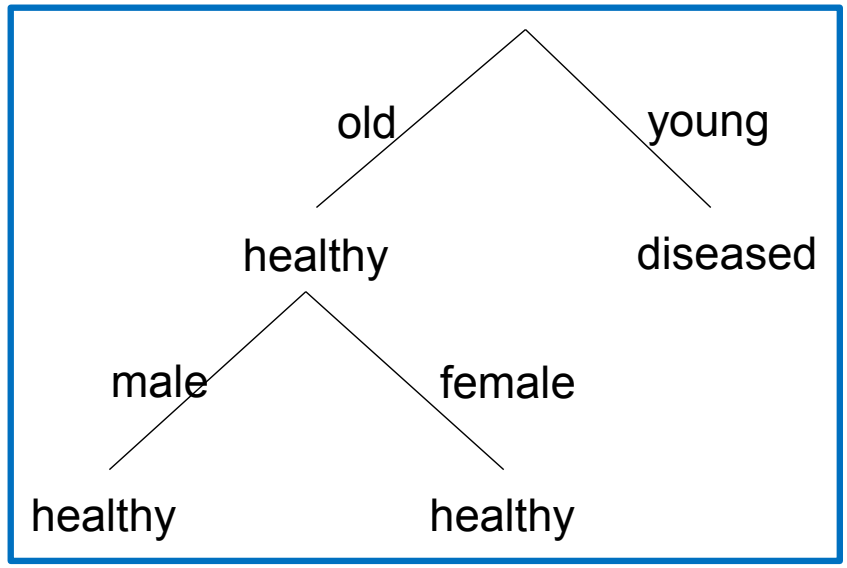
- Draw *B* bootstrap samples from the original dataset
- Grow a tree for each bootstrap sample but
  - For each split of each tree only use a random sample of *m(try)* covariates; *m(try) = sqrt(p) (classn); p/3 (regn)*
    - Bagging precursor has *m(try) = p*
    - Grow to large / maximal depth
    - No pruning / C-V selection to ensure *id*
- Creates a diverse (decorrelated?) collection of trees
  - trees are unstable wrt changes in learning data
  - splits based on random covariate subsets

Intuition of Random Forest

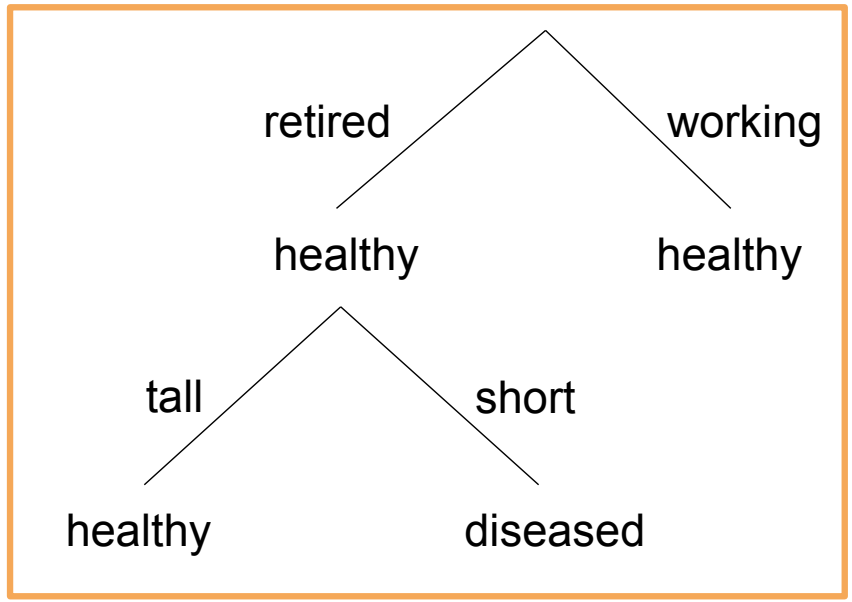
Tree 1



Tree 2



Tree 3



New sample:

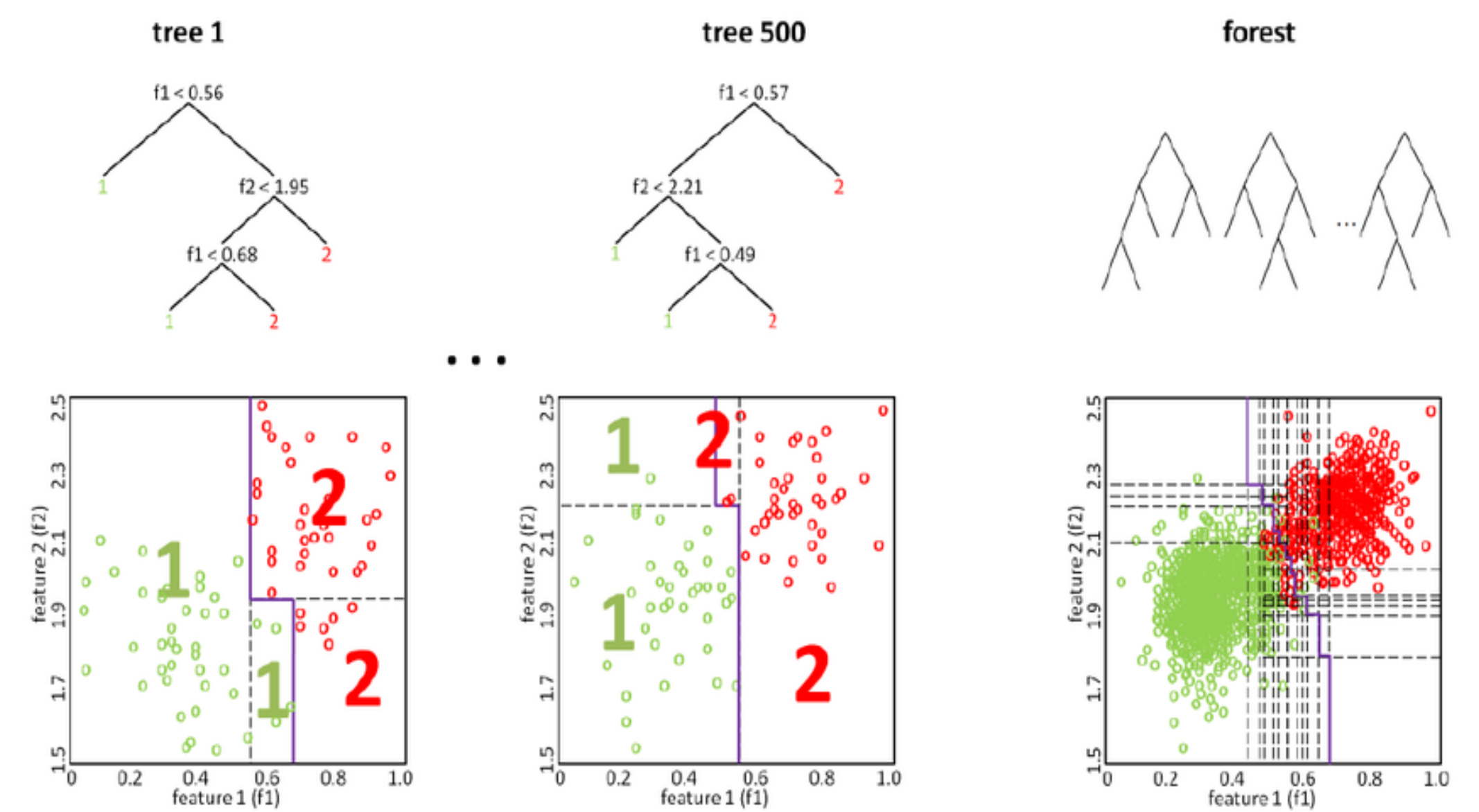
old, retired, male, short

Tree predictions:

diseased, healthy, diseased

Majority rule:

diseased


tree 1
f1 < 0.56
1
f2 < 1.95
f1 < 0.68
2
1
2
tree 500
f1 < 0.57
f2 < 2.21
2
1
f1 < 0.49
1
2
forest
...
feature 2 (f2)
feature 1 (f1)
1
2
1
2
1
2

Random Forest Intuition

- At each split of each tree majority of covariates excluded
  - seems misguided (!)
- Suppose there is a very strong predictor and several moderately strong predictors
- Under bagging, most trees will utilize the strong predictor as first split
  - collection of trees is highly similar
  - predictions from the bagged trees are highly correlated
  - limited gain from averaging highly correlated quantities

$$Var\left(\frac{1}{B}\sum_{i=1}^{B} T_i(c)\right) = \frac{1}{B^2}\sum_{i=1}^{B}\sum_{j=1}^{B} Cov(T_i(x), T_j(x))$$

$$= \frac{1}{B^2}\sum_{i=1}^{B}\left(\sum_{j\neq i}^{B} Cov(T_i(x), T_j(x)) + Var(T_i(x))\right)$$

$$= \frac{1}{B^2}\sum_{i=1}^{B}\left((B-1)\sigma^2\cdot\rho + \sigma^2\right)$$

$$= \frac{B(B-1)\rho\sigma^2 + B\sigma^2}{B^2}$$

$$= \frac{(B-1)\rho\sigma^2}{B} + \frac{\sigma^2}{B}$$

$$= \rho\sigma^2 - \frac{\rho\sigma^2}{B} + \frac{\sigma^2}{B}$$

$$= \rho\sigma^2 + \sigma^2\frac{1-\rho}{B}$$

i=j

Decreaes, if ρ decreases, i.e., if m decreases

De-correlation gives better accuracy

Decreases, if number of trees B increases (irrespective of ρ)

A RF is a collection of tree predictors
$h(\mathbf{x};\boldsymbol{\theta}_t), t = 1,\ldots,T;\quad \boldsymbol{\theta}_t$ *iid* random vectors
For regression, the forest prediction is the unweighted average over the collection: $\bar{h}(\mathbf{x})$

As $t \to \infty$ the Law of Large Numbers ensures
$E_{\mathbf{X},Y}(Y - \bar{h}(X))^2 \to E_{\mathbf{X},Y}(Y - E_{\boldsymbol{\theta}} h(\mathbf{X};\boldsymbol{\theta}))^2$
$\equiv PE_f^*$ the forest prediction error

Convergence implies forests *don't* overfit

Average prediction error for a single tree is

$$PE_t^* = E_{\boldsymbol{\theta}} E_{\mathbf{X},Y} (Y - h(\mathbf{X}; \boldsymbol{\theta}))^2$$

Assume $EY = E_{\mathbf{X}} h(\mathbf{x}; \boldsymbol{\theta}) \ \forall \boldsymbol{\theta}$

Then $PE_f^* \leq \bar{\rho} PE_t^*$ where $\bar{\rho}$ is weighted corr^n between residuals for independent $\boldsymbol{\theta}^{'}, \boldsymbol{\theta}^{''}$

Inequality pinpoints needs for accurate RF: low residual corr^n; low PE for individual trees

Low corr^n sought via injected randomization

But what about low PE_t^*?

- Growing trees to maximal depth minimizes bias
  - But potentially incurs prediction variance cost
  - Averaging over ensemble putatively handles this
- But how was it established that such averaging (more than) compensates for increased individual tree variability??
  - Hard to address theoretically
- Breiman (2001a,b) addressed empirically using
  - UCI Irvine machine learning benchmark datasets
    - Includes classification and regression problems
    - Simulated and (predominantly) real data
    - Exported to R mlbench library

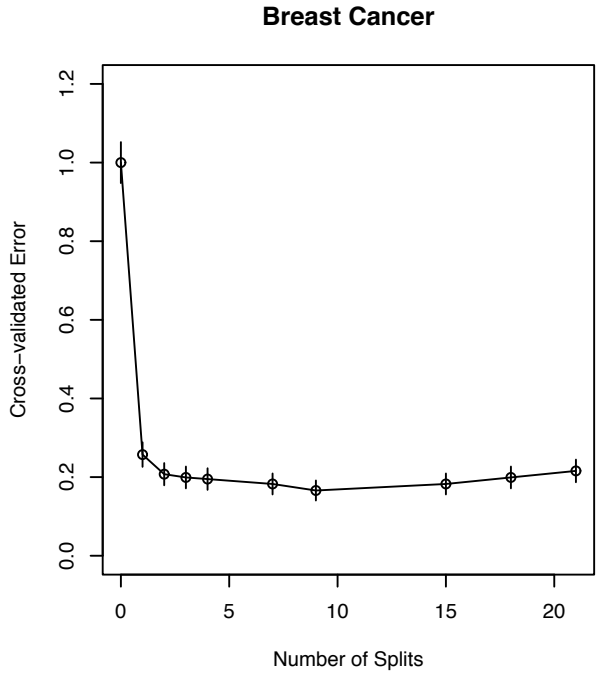
Some classification results from UCI Irvine machine learning benchmark datasets:

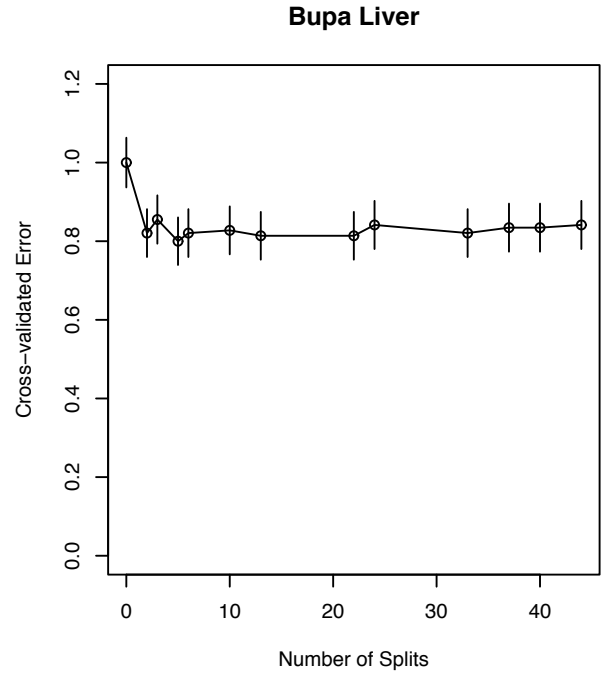
Test set misclassification error (%)

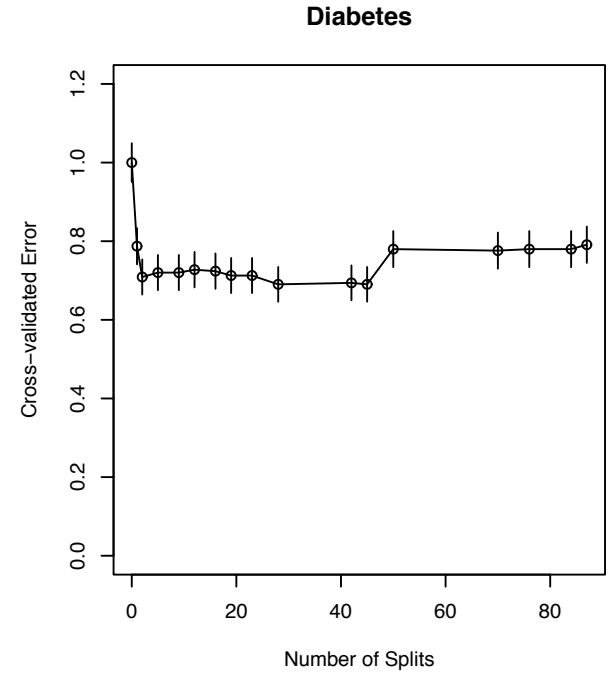
Data set	**Forest**	**Single tree**
Breast cancer	2.9	5.9
Ionosphere	5.5	11.2
Diabetes	24.2	25.3
Glass	22.0	30.4
Soybean	5.7	8.6
Letters	3.4	12.4
Satellite	8.6	14.8
Shuttle $\times 10^3$	7.0	62.0
DNA	3.9	6.2
Digit	6.2	17.1

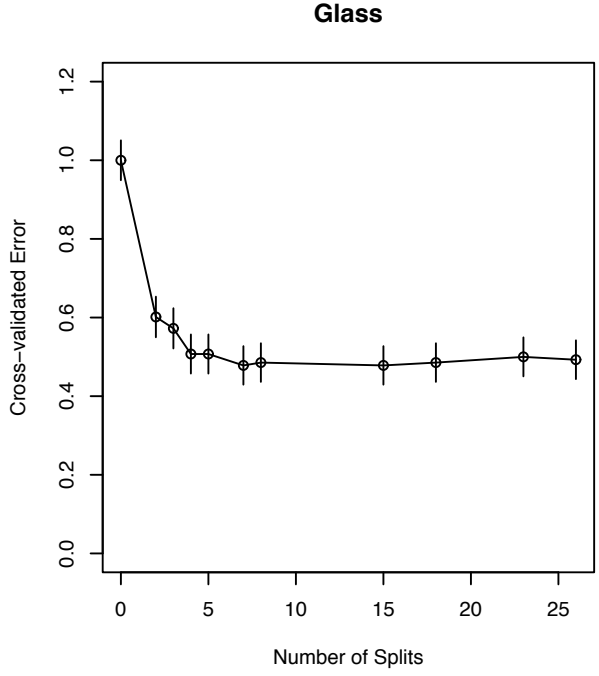
Breiman (2001a,b)

- Many further comparisons using the UCI Irvine / mlbench repository datasets:
  - several modeling / prediction frameworks:
    - CART, ANNs, LDA, QDA, kNNs...
  - regression and classification problems
- Conclusion: "Random Forests are A+ predictors"
- Discussion (Efron): Lots of knobs (tuning parameters)
- Rejoinder (Breiman): Essentially only one (*mtry*)
  - But, lets take a closer look at the UCI Irvine / mlbench repository datasets


Breast Cancer
Cross−validated Error
Number of Splits
0.0
0.2
0.4
0.6
0.8
1.0
1.2
0
5
10
15
20


Bupa Liver
Cross−validated Error
Number of Splits
0.0
0.2
0.4
0.6
0.8
1.0
1.2
0
10
20
30
40


Diabetes
Cross−validated Error
Number of Splits
0.0
0.2
0.4
0.6
0.8
1.0
1.2
0
20
40
60
80


Glass
Cross−validated Error
Number of Splits
0.0
0.2
0.4
0.6
0.8
1.0
1.2
0
5
10
15
20
25

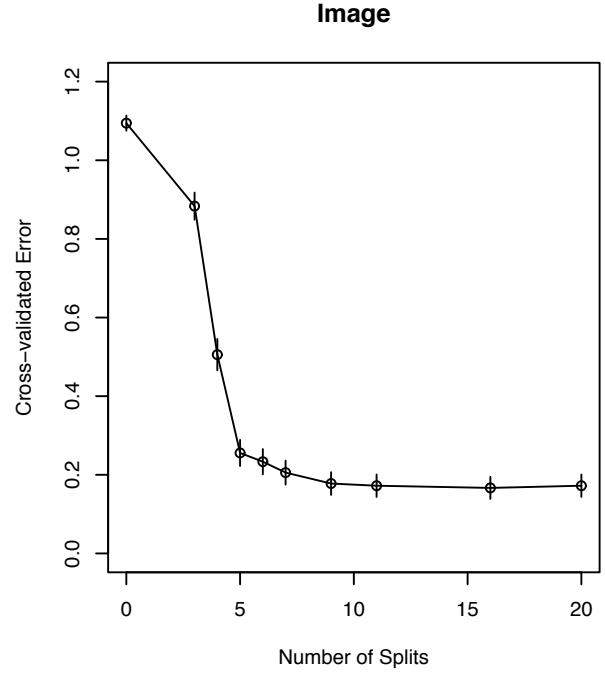
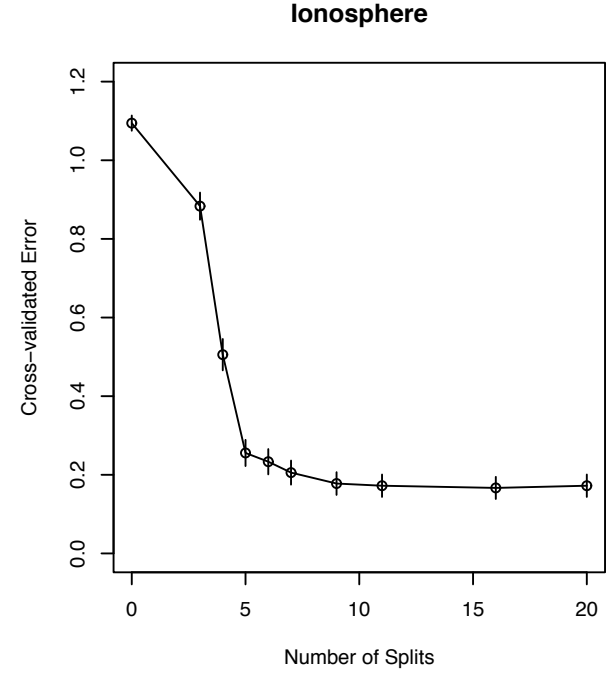
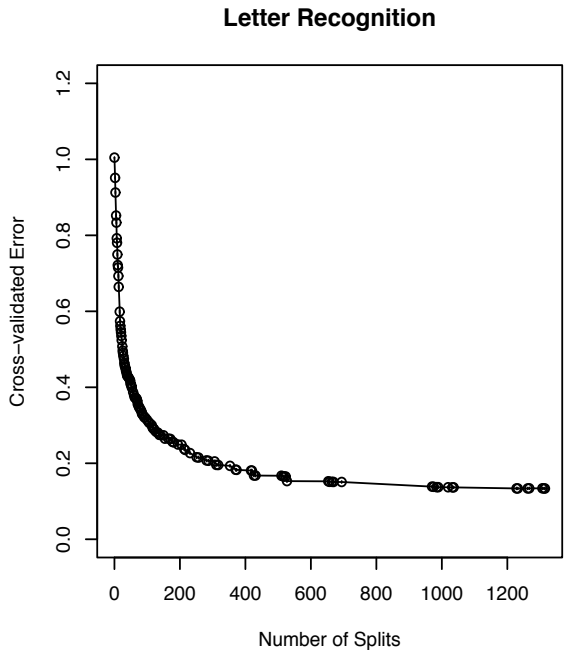
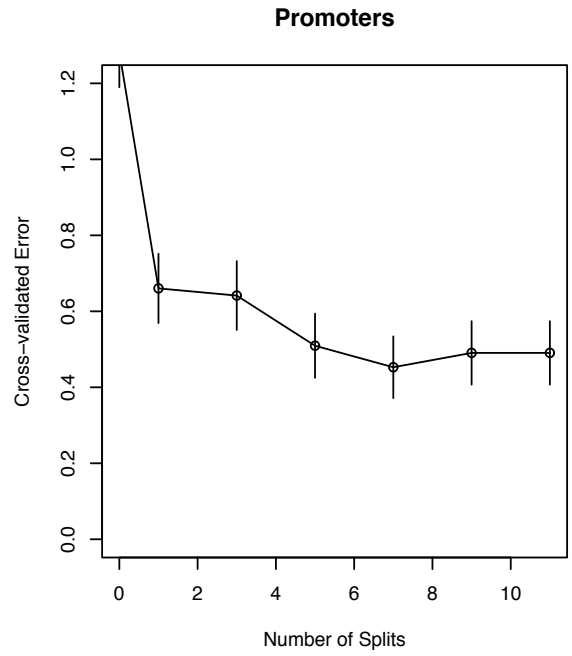
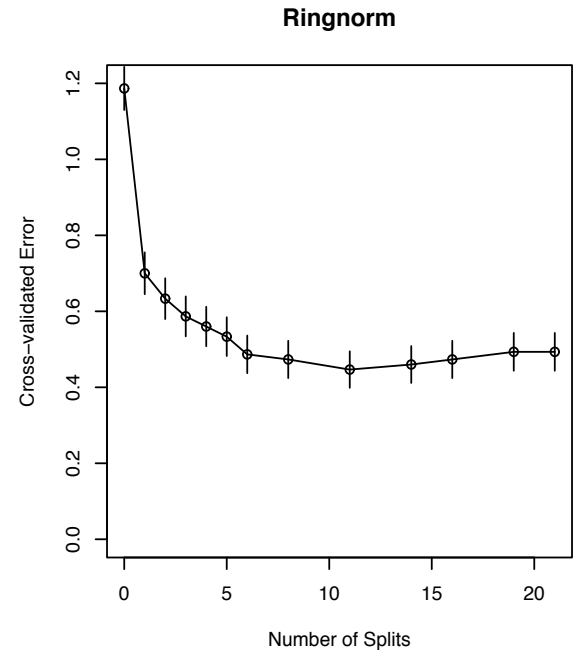


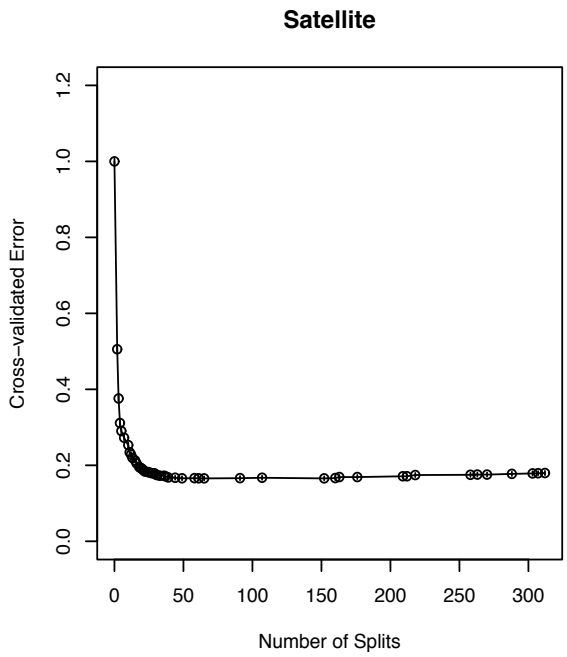
Image
Cross−validated Error
Number of Splits
0.0
0.2
0.4
0.6
0.8
1.0
1.2
0
5
10
15
20

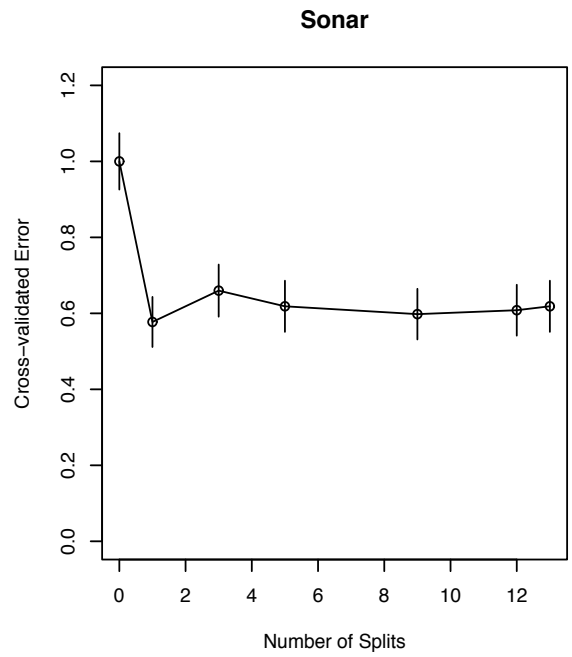

Ionosphere
Cross−validated Error
Number of Splits
0.0
0.2
0.4
0.6
0.8
1.0
1.2
0
5
10
15
20

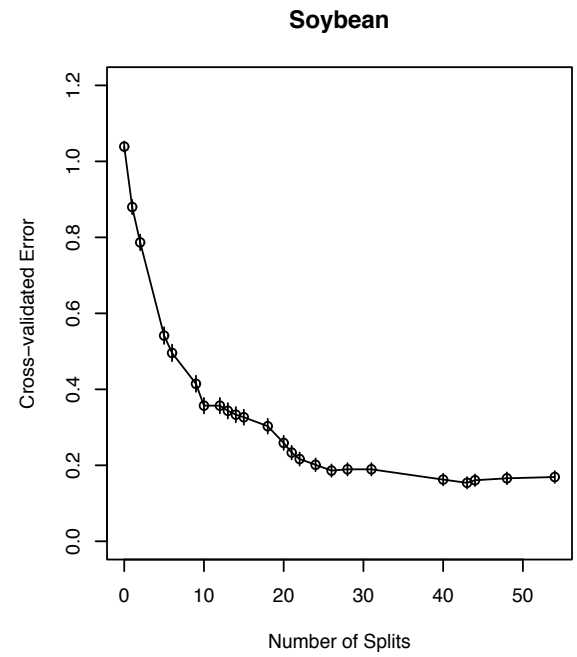

Letter Recognition
Cross−validated Error
Number of Splits
0.0
0.2
0.4
0.6
0.8
1.0
1.2
0
200
400
600
800
1000
1200

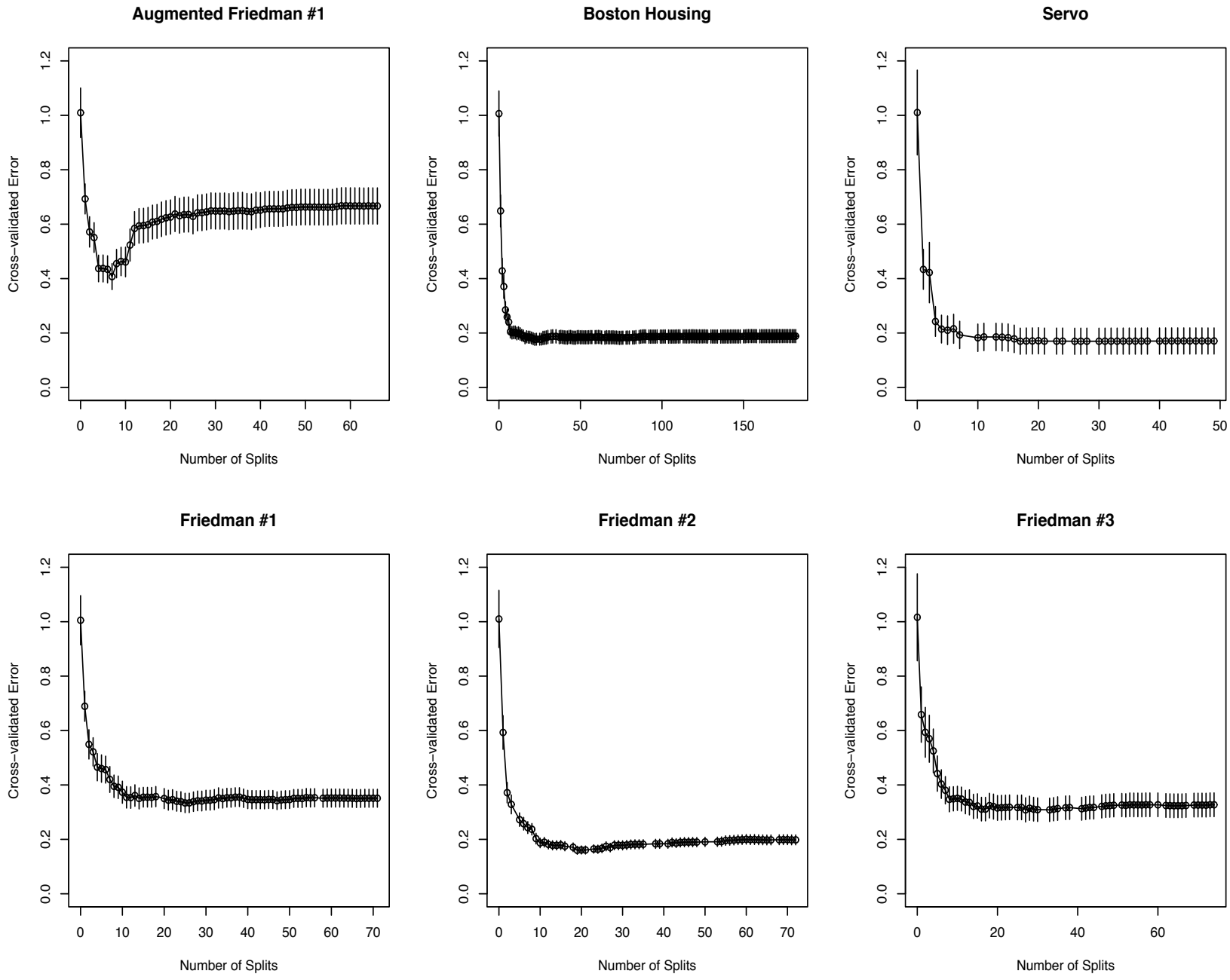

Promoters
Cross−validated Error
Number of Splits
0.0
0.2
0.4
0.6
0.8
1.0
1.2
0
2
4
6
8
10


Ringnorm
Cross−validated Error
Number of Splits
0.0
0.2
0.4
0.6
0.8
1.0
1.2
0
5
10
15
20


Satellite
Cross−validated Error
Number of Splits
0.0
0.2
0.4
0.6
0.8
1.0
1.2
0
50
100
150
200
250
300


Sonar
Cross−validated Error
Number of Splits
0.0
0.2
0.4
0.6
0.8
1.0
1.2
0
2
4
6
8
10
12


Soybean
Cross−validated Error
Number of Splits
0.0
0.2
0.4
0.6
0.8
1.0
1.2
0
10
20
30
40
50


Augmented Friedman #1
Boston Housing
Servo
Friedman #1
Friedman #2
Friedman #3
Cross−validated Error
Number of Splits

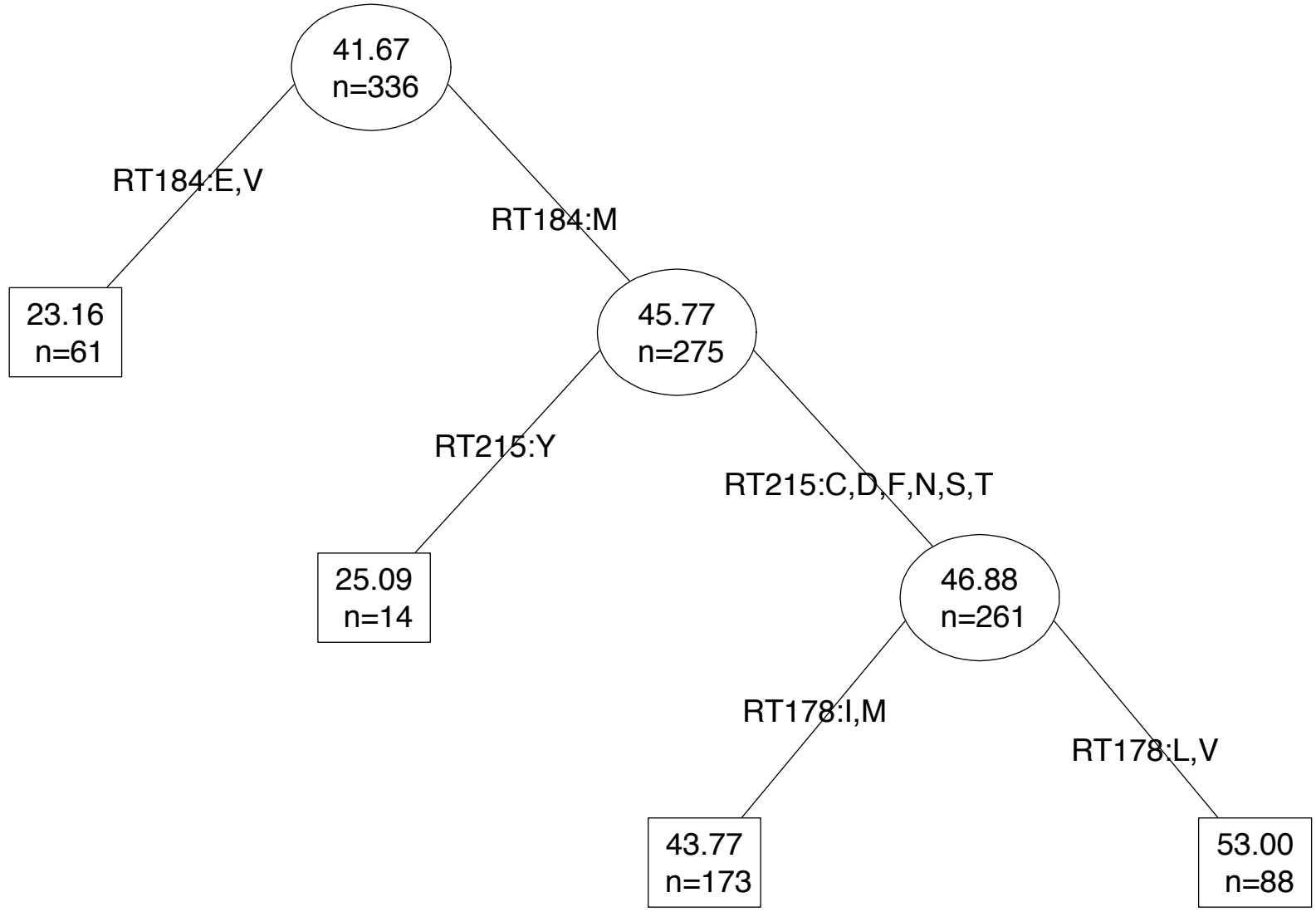
- Almost all UCI Irvine machine learning benchmark datasets exhibit this behavior:
  - they are hard to *overfit* {not just with trees}
- This will make the Random Forest strategy of growing trees to maximal depth look good
- "Benchmarks" are not representative of what is at least thought to be prototypic
  - Will next showcase such an example

HIV-1 Replication Capacity

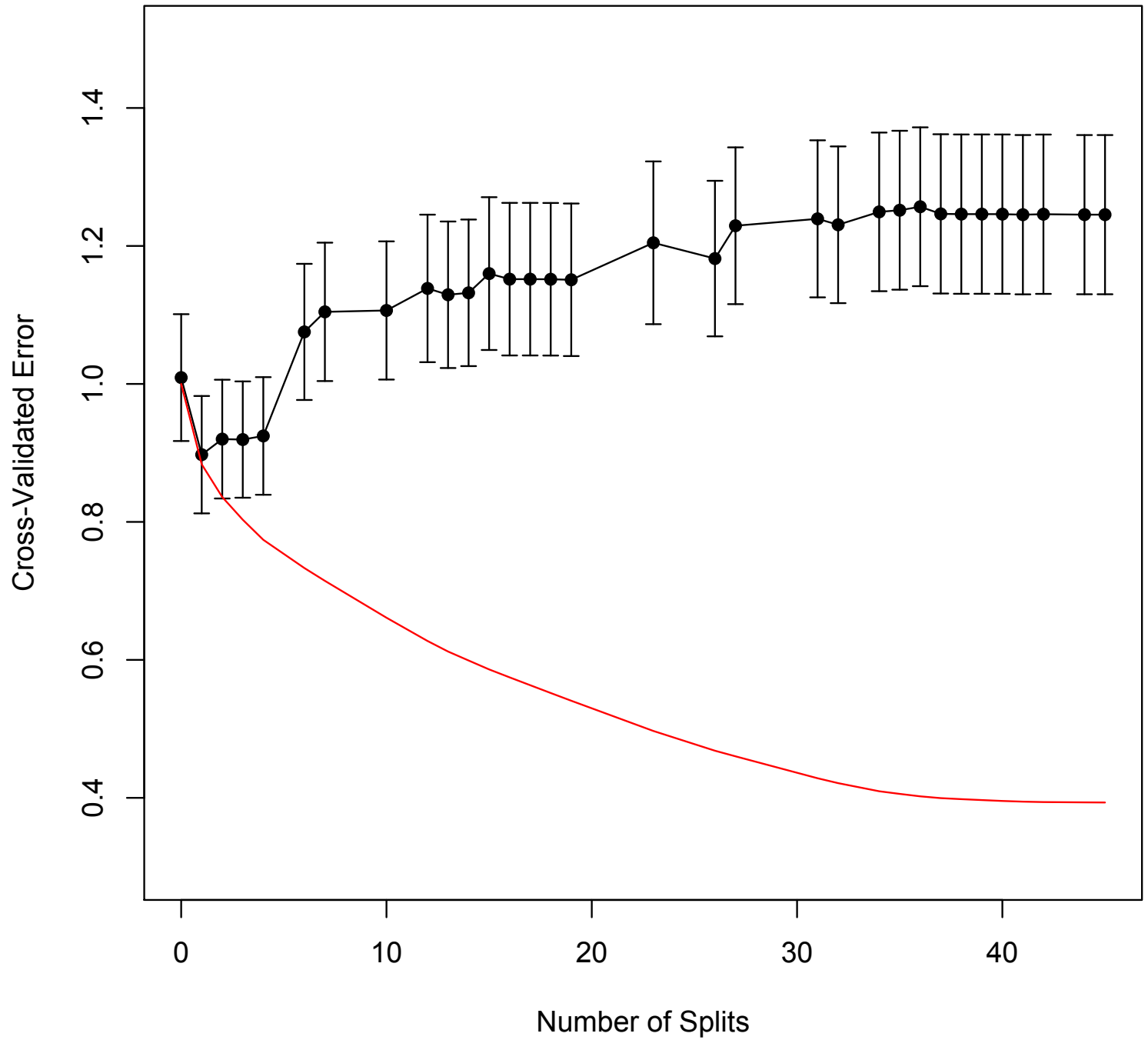
- RC assay measures viral fitness under ideal conditions
  - wide natural variation; unknown genetic basis
  - identification of sites influencing variation important:
    - viral lifecycle, relation to clinical outcomes, virulence, drug targets
- Major anti-retrovirals target reverse transcriptase (RT) or protease (PRO)
  - resistance causing mutations found within these proteins

HIV-1 Replication Capacity

- 336 patients with RC values linked to HIV-1 sequence:
  - PRO codons 4 - 99; RT codons 38 - 223
  - amino acids represent an unordered categorical covariate with 20 levels
    - naturally and exhaustively handled via tree methods
    - smoothness / linearity moot
  - strong pairwise dependencies (Bickel et al., 1996)


41.67
n=336
RT184:E,V
RT184:M
23.16
n=61
45.77
n=275
RT215:Y
RT215:C,D,F,N,S,T
25.09
n=14
46.88
n=261
RT178:I,M
RT178:L,V
43.77
n=173
53.00
n=88


RT185,186
RT177,178


Cross-Validated Error
0.4
0.6
0.8
1.0
1.2
1.4
0
10
20
30
40
Number of Splits

# Splits per Tree	Minimum Node Size	# Covariates per Split (m)			
		10	20	100	276
Unrestricted	5	589.7	590.4	608.2	602.9
	25	589.2	586.7	587.5	593.8
	50	594.0	583.7	582.1	584.2
5	5	602.9	592.9	575.6	578.6
	25	598.5	587.4	576.2	577.1
	50	592.4	588.4	581.2	581.6

Prediction Errors:

Worst performance at default settings

Improvements realized by restricting # Splits per Tree

No gain over individual tree due to strong covariate dependencies

Not overly sensitive to number of trees in the forest provided this is sufficiently large (default 500)

Random Forests

- These examples are cautionary
  - RF remain a forefront prediction technique
- In addition to (generally) improving the prediction performance of individual trees (their main deficiency) RF inherit the other advantages of tree methods:
  - handling missing data, mixed variable types
  - computationally efficient even in large problems
    - speed-ups, parallelization, OOB error estimates
- Interpretation remains an issue as complexities surround widely used variable importance measures

R Software & Extensions

- Tree-structured methods:
  - rpart, party(kit)
- Forests:
  - randomForest, randomForestSRC, ranger
- Multivariate response:
  - integratedMRF
- Stable high-order interactions:
  - iteratedRF