CART
Bagging Trees
Random Forests

Leo Breiman

- Breiman, L., J. Friedman, R. Olshen, and C. Stone, 1984: Classification and regression trees. Wadsworth Books, 358.
- Breiman, L., 1996: Bagging predictors. Machine learning, 24 (2), 123--140.
- Breiman, Leo (2001). "Random Forests". *Machine Learning* **45** (1): 5–32. doi:10.1023/A:1010933404324

Outline

- Regression Tree / Classification Tree
- Snow Example
- Pruning, Cross-Validation
- Example in R
- Compare to linear regression
- Bagging
- Random forests
- Another Example in R

Regression Tree Algorithm

- Let the data be a set of O vector observations, each of length V, such that each observation has one response variable and V-1 predictor variables (supervised learning)

$$o_i = \{o_{i1}, ..., o_{iV}\} = \{r_i, p_{i1}, ..., p_{i(V-1)}\}$$

- 1. For all V-1 predictors,
  - order it's values (separate into categories)
  - partition the sorted predictor variables at every delta in the sorted values (or by excluding any category)
  - partition the associated response variable in the same way and compute its resulting variance (over two groups)
- 2. Choose the partition which minimizes the response variance over all predictors and thresholds.
- 3. Split the data into 2 pieces on this threshold and repeat steps 1 and 2 on both until some stopping rule is satisfied or each partition contains only 1 data point.

Classification scatter / variance / impurity

Misclassification error: $\frac{1}{N_m} \sum_{i \in R_m} I(y_i \neq k(m)) = 1 - \hat{p}_{mk(m)}.$

Gini index: $\sum_{k \neq k'} \hat{p}_{mk} \hat{p}_{mk'} = \sum_{k=1}^{K} \hat{p}_{mk}(1 - \hat{p}_{mk}).$

Cross-entropy or deviance: $-\sum_{k=1}^{K} \hat{p}_{mk} \log \hat{p}_{mk}.$

(9.17)

(Hastie et al, ch 9.2, p. 309)

Terminal Nodes and New Data

- Regression: the mean value of all points in each terminal node is the representative of that terminal node. (variance?)
- Classification: the most popular class in the node is selected.
- Estimation and Prediction: New observation vectors are “dropped down” the tree and are filtered into and end node and its associated response is assigned that value.

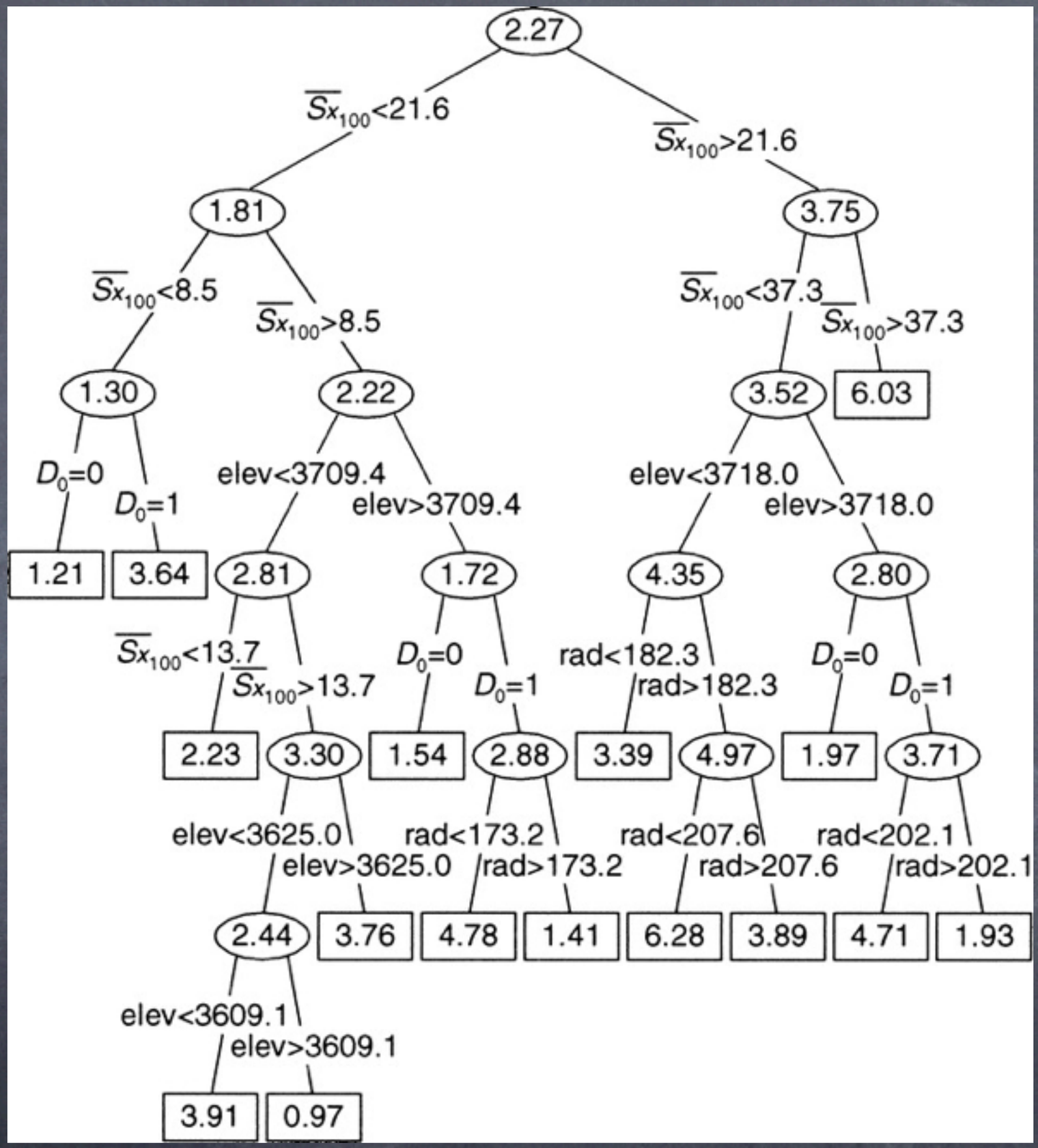

Fig. 10. Pruned 16-node regression tree grown on Sx_{100} (°), D_0 (dimensionless), elevation (m), net potential radiation index (W m^{-2}), and slope. None of the splits were based on slope. Values within the ellipses and rectangles (terminal nodes) are the mean depth (m) of all samples falling within that node. (From: Winstral, Adam, Kelly Elder, Robert E. Davis, 2002: Spatial Snow Modeling of Wind-Redistributed Snow Using Terrain-Based Parameters. *J. Hydrometeor*, **3**, 524–538. doi: 10.1175/1525-7541)

Complete Binary Tree

2^0 1
2^1 2 3
2^2 4 5 6 7
2^3 8 9 10 11 12 13 14 15
2^4 ....

- node 1 is all of the data with full initial scatter/variance in response
- terminal nodes = leaves
- each node n has children 2n and 2n+1
- each level has nodes 2^L - (2^(L+1)-1)

Tree fitting and pruning

- how to find the appropriate level of tree fit??
- over grow tree: over-fitting will cause misclassification
- generate error measures from 10 fold cross-validation
- prune back to terminal node via nodes of minimum loss of error
- compute deviance at each node
- repeat 10-fold CV over some number of runs (100)
- 1-se rule: choose node whose standard error drops below the minimum over all nodes
- high variance of trees - addressed by emsemble methods

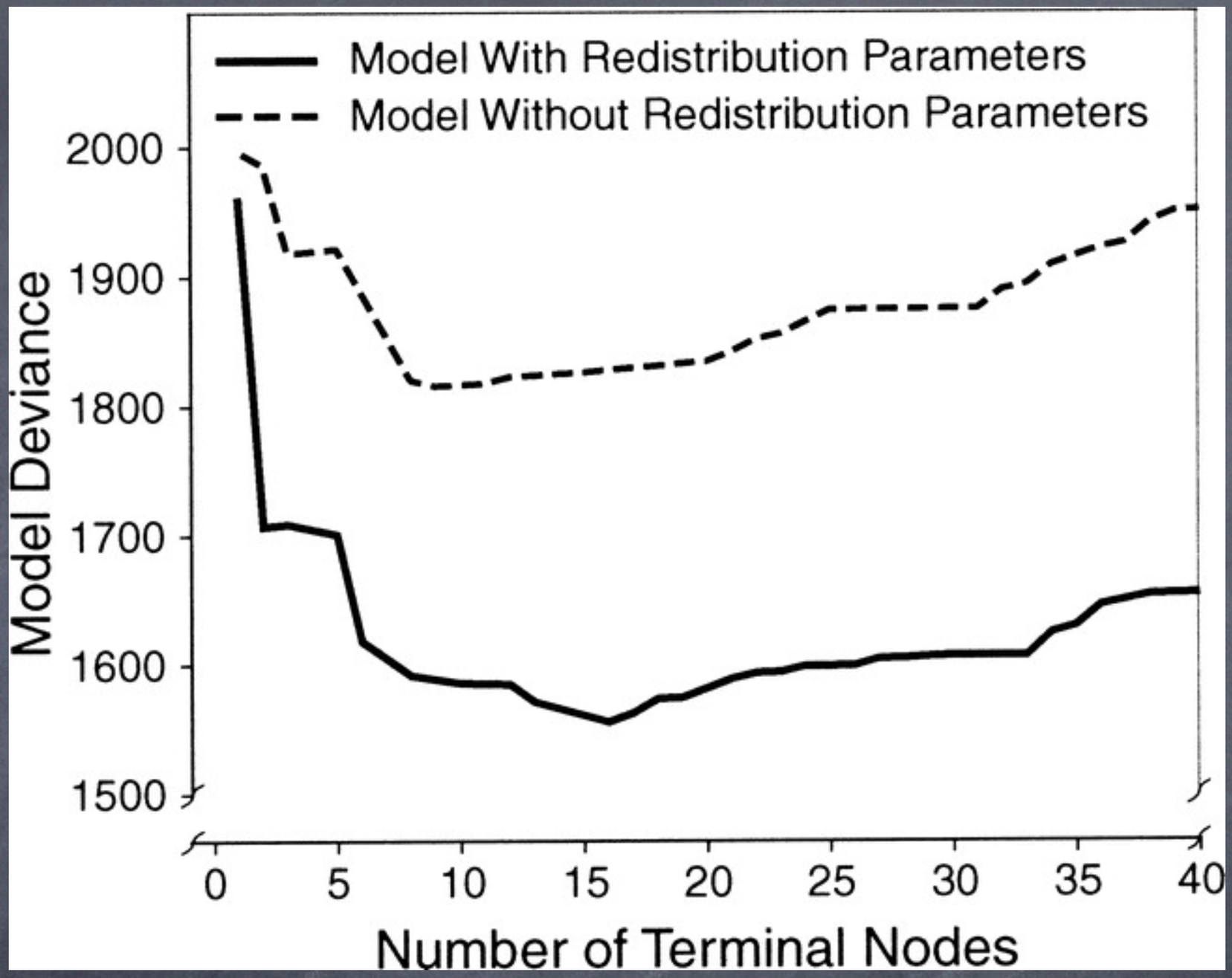

Fig. 9. Cross-validation results for the regression tree models. Suggested tree sizes based on the flat minimums of the plots suggested an optimal tree size of 16 nodes for the redistribution model and a range of 8–20 nodes for the nonredistribution model (From: Winstral et al, 2002.)

Example in R: http://www.statmethods.net/advstats/cart.html

Compare to linear regression

- Has a finite set of possible responses
- May not extrapolate beyond the range of the observed response
- Hierarchical relationship between predictors is unique
- Abrupt vs smooth variation with predictors
- Interesting options for dealing with missing data (see Hastie et al.)

Bagging trees

- Bootstrap AGGregatING trees
- ipred package in R.
- ensemble of trees with different “initial conditions”
- algorithm: in / out of “bag” samples:
  - 1. out of bag: sample from data with replacement (on average get 64% of the full data set)
  - 2. fit a full regression tree (no pruning!)
  - 3. calculate cross-validated stats on in-bag samples
  - repeat 1-3 for some number of trees, nTrees, generating cross validated stats as you go (ideally stats for each data point).
- CV score as a function of the number of trees can decide nTrees
- new data: run down all trees and average their collective result (classification is popular vote)
- bagging is more robust to noise and outliers: the variance of single trees is reduced by their consensus over diverse subsets of the data

Random Forests

- Breiman's web page is a good resource:
  http://www.stat.berkeley.edu/~breiman/RandomForests/
- randomForests package in R, good intro article in R News
  http://cran.r-project.org/doc/Rnews/Rnews_2002-3.pdf
- "Multi model" ensembles

Random Forests Algorithm

- Identical to bagging in every way, except:
  - each time a tree is fit, at each node, censor some of the predictor variables. The number to keep is termed mTry
- 2 parameters: mTry and nTrees

Random Forests Bonuses

- Variable importance
  - scramble each predictor relative to the observations and see if it matters
- proximity of observations
  - how often pairs of observations fall into the same terminal nodes over the forest
  - used for classification where a predictor variable is synthesized

```r
# Regression Tree Example (from the quickR site http://www.statmethods.net/advstats/cart.html)
library(rpart)

# grow tree
fit <- rpart(Mileage~Price + Country + Reliability + Type,
   method="anova", data=cu.summary)

printcp(fit) # display the results
plotcp(fit) # visualize cross-validation results
summary(fit) # detailed summary of splits

# create additional plots
par(mfrow=c(1,2)) # two plots on one page
rsq.rpart(fit) # visualize cross-validation results

# plot tree
par(mfrow=c(1,1))
plot(fit, uniform=TRUE, main="Regression Tree for Mileage ")
text(fit, use.n=TRUE, all=TRUE, cex=.8)

# create attractive postcript plot of tree
post(fit, file = "nice_rpart_tree.ps",
         title = "Regression Tree for Mileage ")

# prune the tree
#pfit<- prune(fit, cp=0.01160389) # from cptable
pfit<- prune(fit, cp=0.025441) # from cptable

## plot the pruned tree
plot(pfit, uniform=TRUE, main="Pruned Regression Tree for Mileage")
text(pfit, use.n=TRUE, all=TRUE, cex=.8)
post(pfit, file = "nice_rpart_pruned_tree.ps",
         title = "Pruned Regression Tree for Mileage")
```

```r
library(ipred) ## bagging
require(plyr)  ## __ply with parallelization
options(warn=1) ## cause my R is old and i specify options(warn=2) on startup
require(doMC); registerDoMC(4) ## register multiple cores

err.vs.ntree <- function(n)  ## pass in the number of trees
  bagging( Mileage~Price + Country + Reliability + Type, nbagg=n,
          data=cu.summary, coob=TRUE)$err

ntree <- seq(10,1000,50)
error <- laply( as.list(ntree), err.vs.ntree, .parallel=TRUE )
plot( ntree, error, type='b' )

## check prediction on 1st point
## (the essence of CV that's not oob - do CV with plyr!)
## note a bunch of points have missing mileages...
bag.fit <- bagging(  Mileage~Price + Country + Reliability + Type, nbagg=100,
                   data=cu.summary[-4,], coob=TRUE)
predict( bag.fit, newdata=cu.summary[4,] )
cu.summary$Mileage[4]

########################################
library(randomForest)
rf.fit <- randomForest( Mileage~Price + Country + Reliability + Type,
                        cu.summary[-4,], na.action='na.omit')
## ?na.action
predict( rf.fit, newdata=cu.summary[4,] )

plot(rf.fit) ## could fit randomForests over mTry & possible increase nTrees
importance(rf.fit) ## large increase in impurity/variance means impoprtant
```