Supplemental – Ensemble Methods And Random Forests

Why Does Ensembling Work?

- Suppose there are 25 base classifiers
  - Each classifier has error rate, ε = 0.35
  - Assume classifiers are independent
  - Probability that the ensemble classifier makes a wrong prediction:

$$\sum_{i=13}^{25} \binom{25}{i} \varepsilon^{i} (1-\varepsilon)^{25-i} = 0.06$$

datasciencedojo
data science for everyone

Binomial Theorem on Model Error

$$\sum_{k=n*0.51}^{n} \binom{n}{k} p^k (1-p)^{n-k}$$

n	Number of models in ensemble
k	More than 51% of N. In voting >50% of the vote gets the final verdict
p	The error rate, the probability of being wrong. Compliment of accuracy (1-accuracy)

Example: assume a 3 model ensemble:

Accuracy of each Model	On Paper	In R
.5	$\binom{3}{2} \times .50^2 \times (1-.50)^{3-2} \approx .5$	1-pbinom(1, size=3, prob=0.50) = 0.5
.51	$\binom{3}{2} \times .49^2 \times (1-.49)^{3-2} \approx .48500$	1-pbinom(1, size=3, prob=0.49) = 0.48500
.65	$\binom{3}{2} \times .35^2 \times (1-.35)^{3-2} \approx .28175$	1-pbinom(1, size=3, prob=0.35) = 0.28175

Ensemble amplifies general performance of individual trees.

Adding More Trees

Example: assume all classifiers have an accuracy of 65%

# of Models	On Paper	In R
3	$\binom{3}{2} \times .35^2 \times (1 - .35)^{3-2} \approx 0.28175$	1 - pbinom(1, size=3, prob=0.35) = 0.28175
12	$\binom{12}{7} \times .35^2 \times (1 - .35)^{12-7} \approx 0.21273$	1 - pbinom(7, size=12, prob=0.35) = 0.21273
25	$\binom{25}{13} \times .35^2 \times (1 - .35)^{25-13} \approx 0.06044$	1 - pbinom(12, size=25, prob=0.35) = 0.06045
100	$\binom{100}{51} \times .35^2 \times (1 - .35)^{100-51} \approx 0.00145$	1 - pbinom(49, size=100, prob=0.35) = 0.00145
500	$\binom{500}{251} \times .35^2 \times (1 - .35)^{500-251} \approx 4.3 \times 10^{-13}$	1 - pbinom(249, size=500, prob=0.35) = 4.38e-12

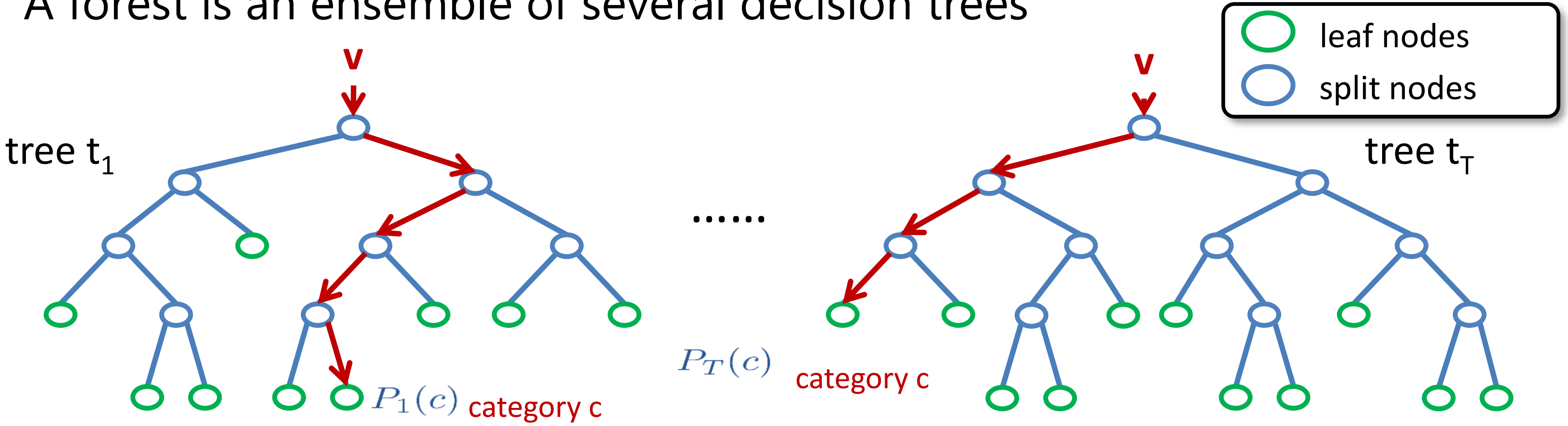
Error rate decreases as the number of (independent) models increases.

Bagging in Random Forests

- A different subset of the training data are selected (~2/3), with replacement, to train each tree
- Remaining training data (aka out-of-bag data or simply OOB) is used to estimate error and variable importance
- Class assignment is made by the number of votes from all of the trees, and for regression, the average of the results is used

A Forest of Trees

A forest is an ensemble of several decision trees



Classification is $P(c|\mathbf{v}) = \frac{1}{T}\sum_{t=1}^{T} P_t(c|\mathbf{v})$

[Amit & Geman 97]
[Breiman 01]
[Lepetit *et al.* 06]

Learning a Forest – GPU Performance

- Dividing training examples into T subsets improves generalization
  - Reduces memory requirements & training time
- Train each decision tree t on subset I_t
  - Same decision tree learning as before
- Multi-core friendly (GPU implementation)

Implementation Details

- How many features and thresholds to try?
  - Just one = “extremely randomized” [Geurts *et al.* 06]
  - Few → fast training, may under-fit, may be too deep
  - Many → slower training, may over-fit
- When to stop growing the tree?
  - Maximum depth
  - Minimum entropy gain
  - Delta class distribution
  - Pruning

Forest Error Rate

$$PE^* \leq \rho(1 - s^2) / s^2$$

- The correlation between any two trees in the forest. Increasing the correlation increases the forest error rate.
- The strength of each individual tree in the forest. A tree with a low error rate is a strong classifier. Increasing the strength of the individual trees decreases the forest error rate.
- Detailed proof: RANDOM FORESTS Leo Breiman

randomForest

R package for Random Forest

Setting up randomForest

Installation

```
> install.packages('randomForest')
```

Loading in R environment

```
> library(randomForest)
```

Documentation:

http://cran.r-project.org/web/packages/randomForest/randomForest.pdf

datasciencedojo
data science for everyone

Obtaining the model

```
> iris.rf <- randomForest(
      Species ~ .,
      data=iris,
      importance=TRUE,
      proximity=TRUE
)
```

Printing the model

```
> print(iris.rf)
```

\# Type of random forest: classification Number of trees: 500

\# No. of variables tried at each split: 2

\# OOB estimate of error rate: 4.67%

\# Confusion matrix: setosa versicolor virginica class.error

# setosa	50	0	0	0.00
# versicolor	0	47	3	0.06
# virginica	0	4	46	0.08

Restricting Tree Size

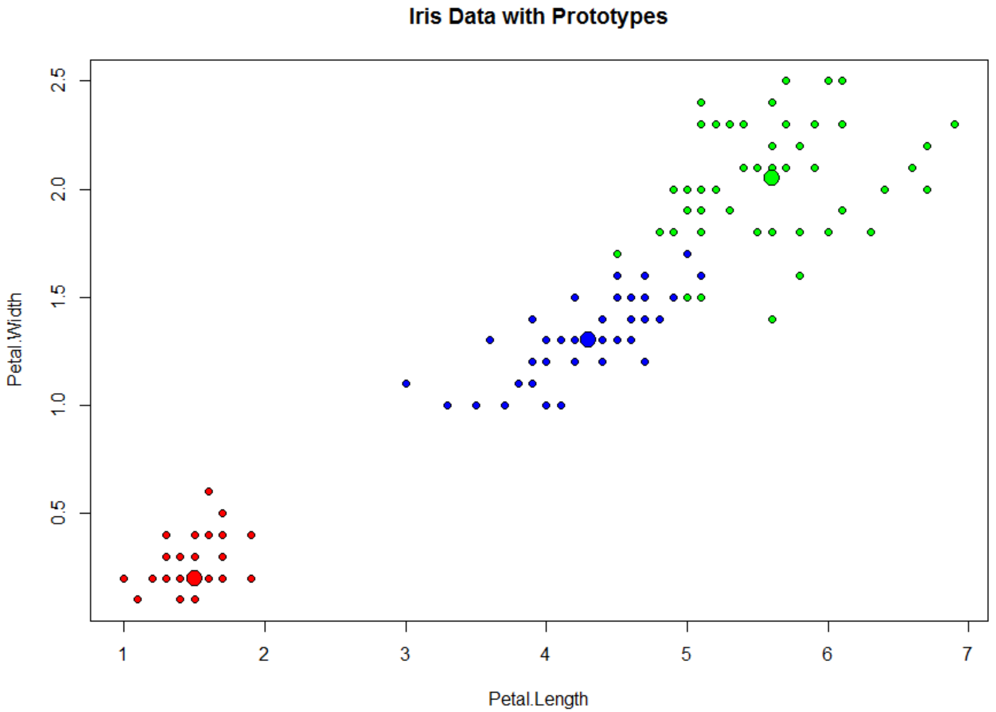
```
(treesize(randomForest(Species ~ .,
          data=iris,
          maxnodes=4,
          ntree=30)))
```

Finding Class Centers

classCenter function

```
> data(iris)
> iris.rf <- randomForest(iris[,-5], iris[,5], prox=TRUE)
> iris.p <- classCenter(iris[,-5], iris[,5], iris.rf$prox)
> plot(iris[,3], iris[,4],
       pch=21, xlab=names(iris)[3],ylab=names(iris)[4],
       bg=c("red","blue","green")[as.numeric(factor(iris$Species))],
       main="Iris Data with Prototypes")
> points(iris.p[,3], iris.p[,4], pch=21, cex=2,
       bg=c("red", "blue", "green"))
```

Finding Class Centers



datasciencedojo
data science for everyone

Combining Trees

You can combine two or more random forests into one.
The confusion, err.rate, mse, and rsq components will be NULL

```
> data(iris)
> rf1 <- randomForest(Species ~ ., iris, ntree=50,
norm.votes=FALSE)
> rf2 <- randomForest(Species ~ ., iris, ntree=100,
norm.votes=FALSE)
> rf3 <- randomForest(Species ~ ., iris, ntree=150,mtry=3,
norm.votes=FALSE)
> rf.all <- combine(rf1, rf2, rf3)
> print(rf.all)
```

Combining Trees

```
> print(rf.all)
```

Call:

randomForest(formula = Species ~ ., data = iris, ntree = 50, norm.votes = FALSE)

Type of random forest:

classification Number of trees: 150

No. of variables tried at each split: 2

Variable Importance and Gini

```
> data(iris)
> iris.rf <- randomForest(Species ~., data=iris, importance=TRUE,
        proximity=TRUE)
> round(importance(iris.rf), 2)
```

	setosa	versicolor	virginica	MeanDecreaseAccuracy	MeanDecreaseGini
Sepal.Length	6.28	8.88	7.11	10.48	9.26
Sepal.Width	4.87	0.77	4.85	5.17	2.33
Petal.Length	21.48	33.88	28.44	33.95	42.97
Petal.Width	22.96	32.47	32.10	34.60	44.65

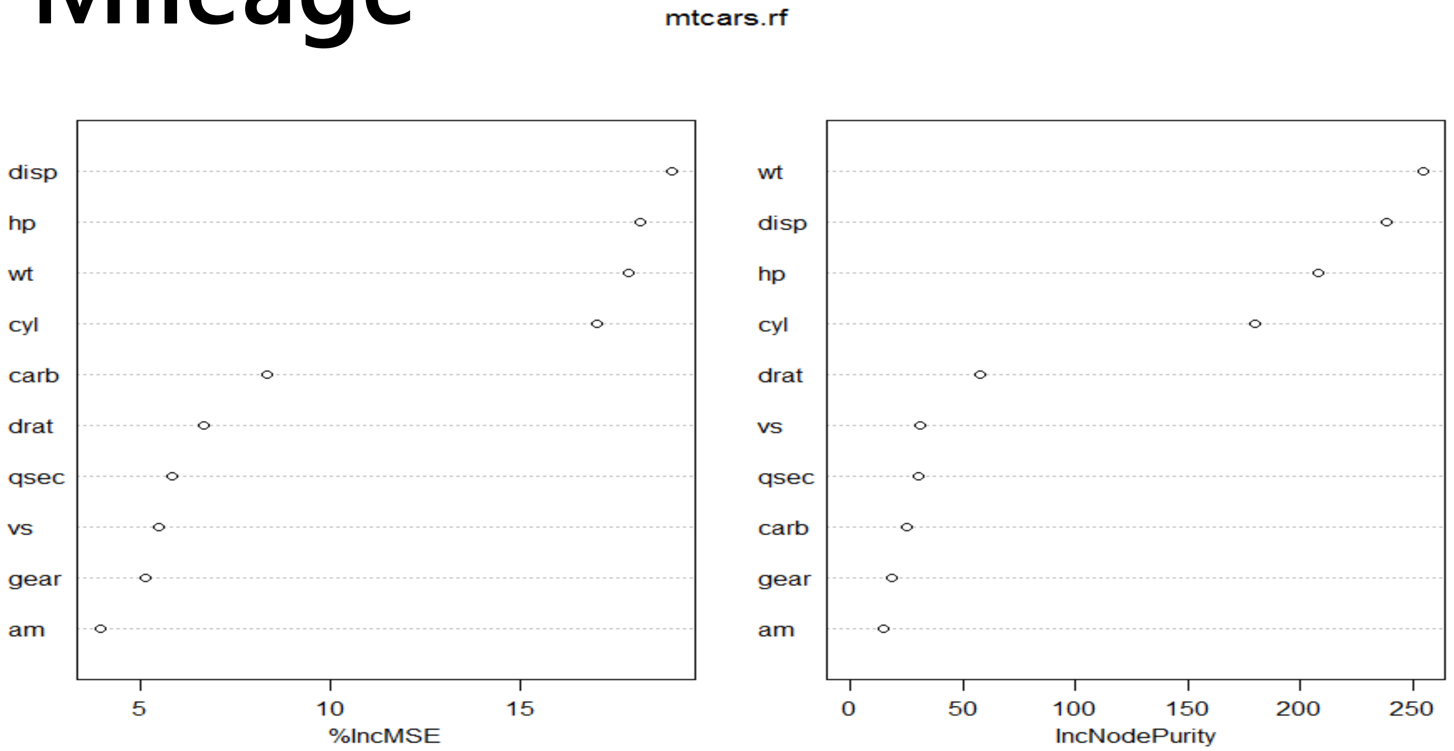
Important Factors In Determining Car Mileage

```
> set.seed(4543)
> data(mtcars)
> mtcars.rf <- randomForest(
    mpg ~ .,
    data=mtcars,
    ntree=1000,
    keep.forest=FALSE,
    importance=TRUE
)
> varImpPlot(mtcars.rf)
```

Index	Attribute Name	Description
[, 1]	mpg	Miles/(US) gallon
[, 2]	cyl	Number of cylinders
[, 3]	disp	Displacement (cu.in.)
[, 4]	hp	Gross horsepower
[, 5]	drat	Rear axle ratio
[, 6]	wt	Weight (lb/1000)
[, 7]	qsec	1/4 mile time
[, 8]	vs	V/S
[, 9]	am	Transmission (0 = automatic, 1 = manual)
[,10]	gear	Number of forward gears
[,11]	carb	Number of carburetors

datasciencedojo
data science for everyone

Important Factors In Determining Car Mileage



Regression With Random Forests

```
> set.seed(131)
> ozone.rf <-
        randomForest(Ozone ~
.,
        data=airquality,
mtry=3,
        importance=TRUE,
        na.action=na.omit)
> print(ozone.rf)
#Impute Missing Values by median/mode.
> ozone.rf <-
        na.roughfix(ozone.rf
)
```

	Ozone	Solar.R	Wind	Temp	Month	Day
1	41	190	7.4	67	5	1
2	36	118	8	72	5	2
3	12	149	12.6	74	5	3
4	18	313	11.5	62	5	4
5	NA	NA	14.3	56	5	5
6	28	NA	14.9	66	5	6
7	23	299	8.6	65	5	7
8	19	99	13.8	59	5	8
9	8	19	20.1	61	5	9
10	NA	194	8.6	69	5	10
11	7	NA	6.9	74	5	11
12	16	256	9.7	69	5	12
13	11	290	9.2	66	5	13
14	14	274	10.9	68	5	14
15	18	65	13.2	58	5	15

Prediction

```
> predict(ozone.rf,data=airquality)
```

Resources

Random Forests Homepage

http://www.stat.berkeley.edu/~breiman/RandomForests/cc_home.htm

IRIS Data

IRIS data ships with R. You can learn more about IRIS data here:

http://archive.ics.uci.edu/ml/datasets/Iris

datasciencedojo
data science for everyone