MACHINE LEARNING WITH TREE-BASED MODELS IN R

Introduction to Random Forest

Erin LeDell
Instructor

Random Forest

- better performance
- sample subset of the features
- improved version of bagging
- reduced correlation between the sampled trees

Random Forest in R

```
> library(randomForest)
```

```
> ?randomForest
```

randomForest {randomForest} R Documentation

Classification and Regression with Random Forest

Description

`randomForest` implements Breiman's random forest algorithm (based on Breiman and Cutler's original Fortran code) for classification and regression. It can also be used in unsupervised mode for assessing proximities among data points.

Usage

```
## S3 method for class 'formula'
randomForest(formula, data=NULL, ..., subset, na.action=na.fail)
## Default S3 method:
randomForest(x, y=NULL,  xtest=NULL, ytest=NULL, ntree=500,
             mtry=if (!is.null(y) && !is.factor(y))
             max(floor(ncol(x)/3), 1) else floor(sqrt(ncol(x))),
             replace=TRUE, classwt=NULL, cutoff, strata,
             sampsize = if (replace) nrow(x) else ceiling(.632*nrow(x)),
             nodesize = if (!is.null(y) && !is.factor(y)) 5 else 1,
             maxnodes = NULL,
             importance=FALSE, localImp=FALSE, nPerm=1,
             proximity, oob.prox=proximity,
             norm.votes=TRUE, do.trace=FALSE,
             keep.forest=!is.null(y) && is.null(xtest), corr.bias=FALSE,
             keep.inbag=FALSE, ...)
## S3 method for class 'randomForest'
print(x, ...)
```

randomForest Example

```
library(randomForest)

# Train a default RF model (500 trees)

model <- randomForest(formula = response ~ ., data = train)
```

MACHINE LEARNING WITH TREE-BASED MODELS IN R

Let's practice!

MACHINE LEARNING WITH TREE-BASED MODELS IN R

Understanding the Random Forest model output

Erin LeDell
Instructor

Random Forest output

```
# Print the credit_model output

> print(credit_model)


Call:
 randomForest(formula = default ~ ., data = credit_train)
               Type of random forest: classification
                     Number of trees: 500
No. of variables tried at each split: 4

        OOB estimate of  error rate: 24.12%
Confusion matrix:
     no yes class.error
no  516  46  0.08185053
yes 147  91  0.61764706
```

Out-of-bag error matrix

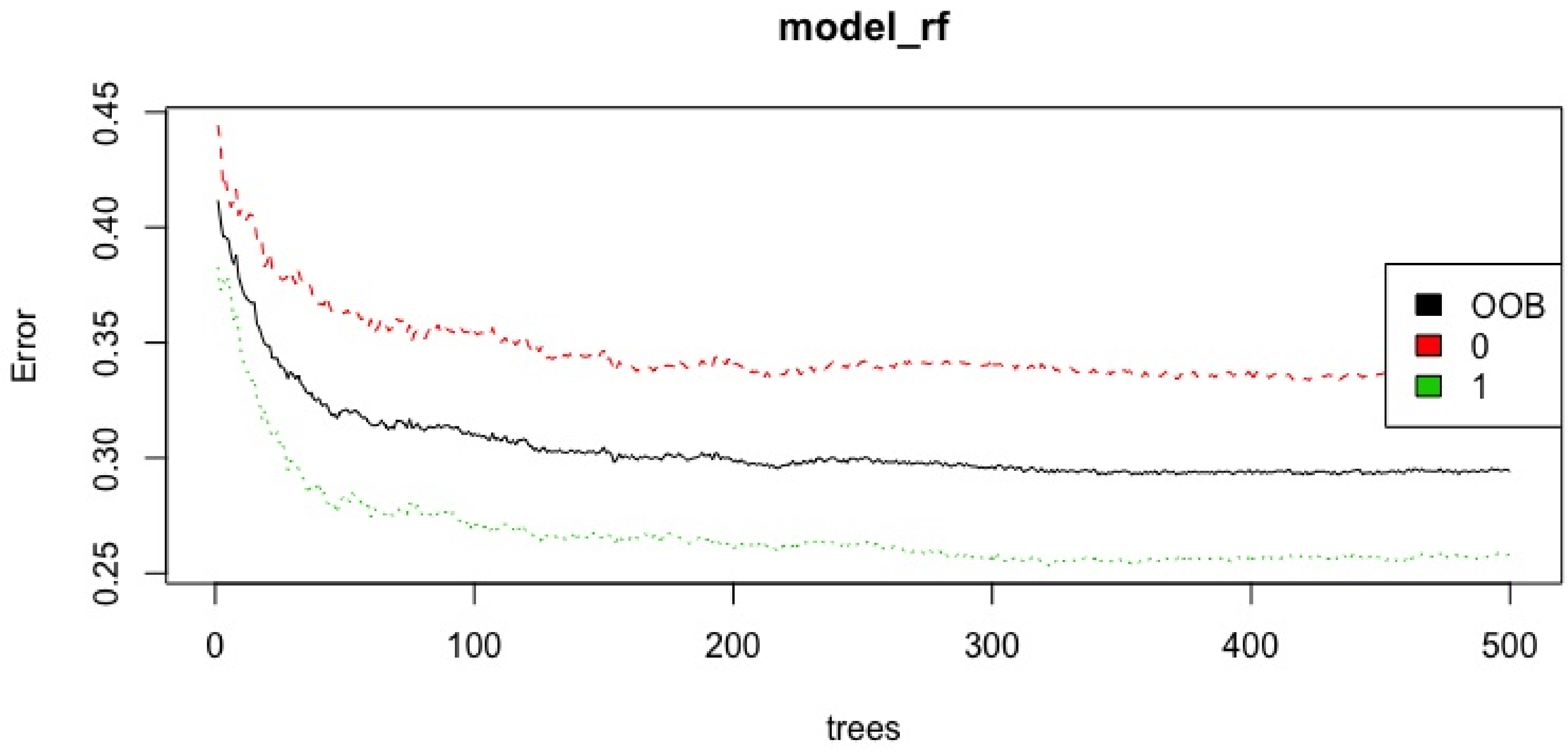
```
# Grab OOB error matrix & take a look

> err <- credit_model$err.rate

> head(err)
           OOB        no       yes
[1,] 0.3414634 0.2657005 0.5375000
[2,] 0.3311966 0.2462908 0.5496183
[3,] 0.3232831 0.2476636 0.5147929
[4,] 0.3164933 0.2180294 0.5561224
[5,] 0.3197756 0.2095808 0.5801887
[6,] 0.3176944 0.2115385 0.5619469
```

Out-of-bag error estimate

```
# Look at final OOB error rate (last row in err matrix)

> oob_err <- err[nrow(err), "OOB"]
> print(oob_err)
   OOB
0.24125
```

```
> print(credit_model)


Call:
 randomForest(formula = default ~ ., data = credit_train)
               Type of random forest: classification
                     Number of trees: 500
No. of variables tried at each split: 4

        OOB estimate of  error rate: 24.12%
Confusion matrix:
     no yes class.error
no  516  46  0.08185053
yes 147  91  0.61764706
```

Plot the OOB error rates



MACHINE LEARNING WITH TREE-BASED MODELS IN R

Let's practice!

MACHINE LEARNING WITH TREE-BASED MODELS IN R

OOB error vs. test set error

Erin LeDell
Instructor

Advantages & Disadvantages of OOB estimates

✔ Can evaluate your model without a separate test set

✔ Computed automatically by the `randomForest()` function

✖ OOB Error only estimates error (not AUC, log-loss, etc.)

✖ Can't compare Random Forest performace to other types of models

MACHINE LEARNING WITH TREE-BASED MODELS IN R

Let's practice!

MACHINE LEARNING WITH TREE-BASED MODELS IN R

Tuning a Random Forest model

Erin LeDell
Instructor

Random Forest Hyperparameters

- ntree: number of trees
- mtry: number of variables randomly sampled as candidates at each split
- sampsize: number of samples to train on
- nodesize: minimum size (number of samples) of the terminal nodes
- maxnodes: maximum number of terminal nodes

Tuning mtry with tuneRF()

```
# Execute the tuning process

set.seed(1)
res <- tuneRF(x = train_predictor_df,
              y = train_response_vector,
              ntreeTry = 500)
```

```
# Look at results
print(res)
```

Results table:

```
      mtry OOBError
2.OOB    2   0.2475
4.OOB    4   0.2475
8.OOB    8   0.2425
```

MACHINE LEARNING WITH TREE-BASED MODELS IN R

Let's practice!