

DATA ANALYTICS AND BUSINESS INTELLIGENCE (8696/8697)

ENSEMBLE DECISION TREES

Graham.Williams@togaware.com

Data Scientist
Australian Taxation Office

Adjunct Professor, University of Canberra

Graham.Williams@togaware.com
<http://datamining.togaware.com>



OVERVIEW

- 1 OVERVIEW
- 2 MULTIPLE MODELS
- 3 BOOSTING
 - Algorithm
 - Example
- 4 RANDOM FORESTS
 - Forests of Trees
 - Introduction
- 5 OTHER ENSEMBLES
 - Ensembles of Different Models



OVERVIEW

- 1 OVERVIEW
- 2 MULTIPLE MODELS**
- 3 BOOSTING
 - Algorithm
 - Example
- 4 RANDOM FORESTS
 - Forests of Trees
 - Introduction
- 5 OTHER ENSEMBLES
 - Ensembles of Different Models

BUILDING MULTIPLE MODELS

- General idea developed in Multiple Inductive Learning algorithm (Williams 1987).
- Ideas were developed (ACJ 1987, PhD 1990) in the context of:
 - observe that variable selection methods don't discriminate;
 - so build multiple decision trees;
 - then combine into a single model.
- Basic idea is that multiple models, like multiple experts, may produce better results when working together, rather than in isolation
- Two approaches covered: **Boosting** and **Random Forests**.
- Meta learners.



OVERVIEW

- 1 OVERVIEW
- 2 MULTIPLE MODELS
- 3 **BOOSTING**
 - Algorithm
 - Example
- 4 RANDOM FORESTS
 - Forests of Trees
 - Introduction
- 5 OTHER ENSEMBLES
 - Ensembles of Different Models



BOOSTING ALGORITHMS

BASIC IDEA: BOOST OBSERVATIONS THAT ARE “HARD TO MODEL.”

Algorithm: iteratively build **weak models** using a poor learner:

- Build an initial model;
- Identify mis-classified cases in the training dataset;
- Boost (over-represent) training observations modelled incorrectly;
- Build a new model on the boosted training dataset;
- Repeat.

The result is an ensemble of weighted models.

Best off the shelf model builder. (Leo Brieman)



ALGORITHM IN PSEUDO CODE

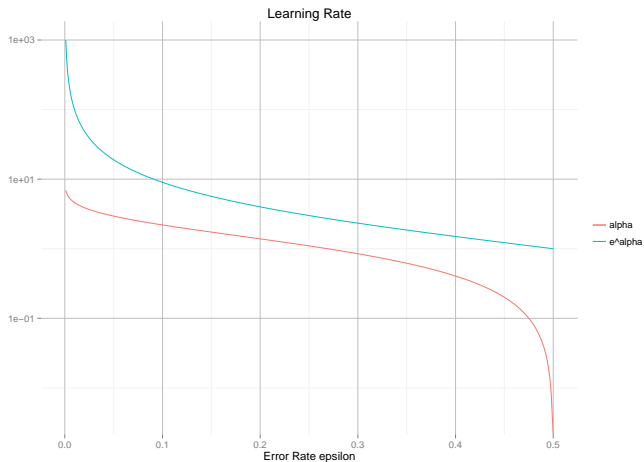
```

adaBoost <- function(form, data, learner)
{
  w <- rep(1/nrows(data), nrows(data))
  e <- NULL
  a <- NULL
  m <- list()
  i <- 0
  repeat
  {
    i <- i + 1
    m <- c(m, learner(form, data, w))
    ms <- which(predict(m[i], data) != data[target(form)])
    e <- c(e, sum(w[ms])/sum(w))
    a <- c(a, log((1-e[i])/e[i]))
    w[ms] <- w[ms] * exp(a[i])
    if (e[i] >= 0.5) break
  }
  return(sum(a * sapply(m, predict, data)))
}

```



DISTRIBUTIONS



EXAMPLE: FIRST ITERATION

```
n <- 10
w <- rep(1/n, n)           # 0.1 0.1 ...
ms <- c(7, 8, 9, 10)
e <- sum(w[ms])/sum(w)     # 0.4
a <- log((1-e)/e)          # 0.4055
w[ms] <- w[ms] * exp(a)    # 0.15 0.15 0.15 0.15
```

EXAMPLE: SECOND ITERATION

```
ms <- c(1, 8)                # 0.10 0.15
w[ms]

## [1] 0.10 0.15

e <- sum(w[ms])/sum(w)       # 0.2083
a <- log((1-e)/e)            # 1.335
(w[ms] <- w[ms] * exp(a))

## [1] 0.38 0.57
```

EXAMPLE: ADA ON WEATHER DATA

```
head(weather[c(1:5, 23, 24)], 3)
```

```
##           Date Location MinTemp MaxTemp Rainfall RISK_MM...
## 1 2007-11-01 Canberra      8.0     24.3        0.0      3.6...
## 2 2007-11-02 Canberra     14.0     26.9         3.6      3.6...
## 3 2007-11-03 Canberra     13.7     23.4         3.6     39.8...
....
```

```
set.seed(42)
```

```
train <- sample(1:nrow(weather), 0.7 * nrow(weather))
(m <- ada(RainTomorrow ~ ., weather[train, -c(1:2, 23)]))
```

```
## Call:
```

```
## ada(RainTomorrow ~ ., data=weather[train, -c(1:2, 23)])
```

```
##
```

```
## Loss: exponential Method: discrete   Iteration: 50
```

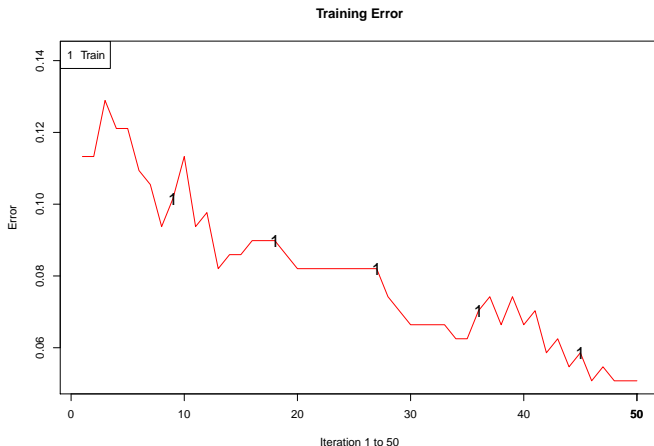
```
....
```



EXAMPLE: ERROR RATE

Notice error rate decreases quickly then flattens.

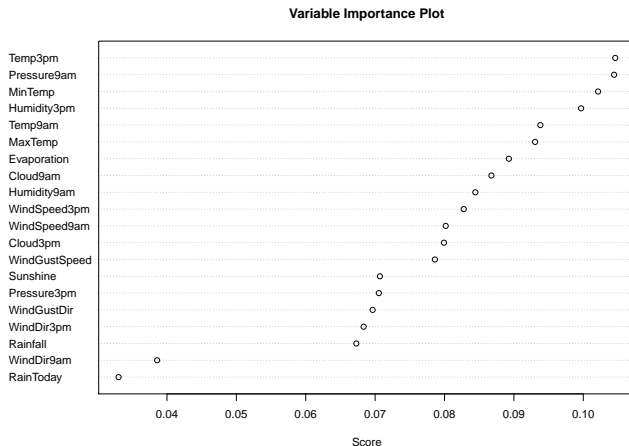
```
plot(m)
```



EXAMPLE: VARIABLE IMPORTANCE

Helps understand the *knowledge* captured.

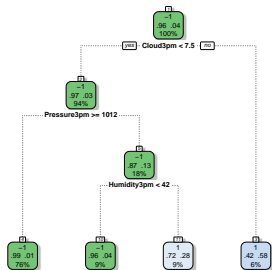
`varplot(m)`



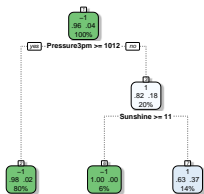
EXAMPLE: SAMPLE TREES

There are 50 trees in all. Here's the first 3.

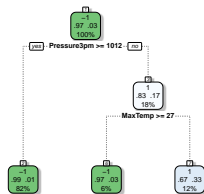
```
fancyRpartPlot(m$model$trees[[1]])
fancyRpartPlot(m$model$trees[[2]])
fancyRpartPlot(m$model$trees[[3]])
```



Rattle 2014-Jul-01 20:15:55 gjw



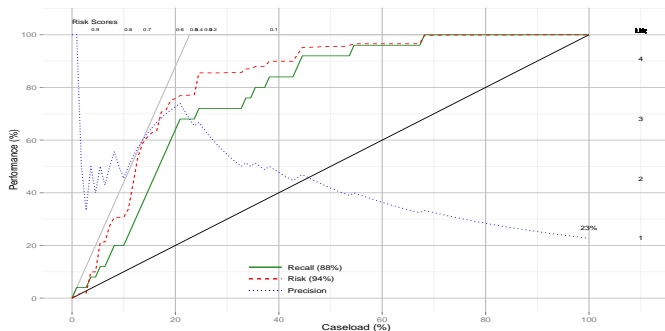
Rattle 2014-Jul-01 20:15:55 gjw



Rattle 2014-Jul-01 20:15:55 gjw

EXAMPLE: PERFORMANCE

```
predicted <- predict(m, weather[-train,], type="prob")[,2]
actual <- weather[-train,]$RainTomorrow
risks <- weather[-train,]$RISK_MM
riskchart(predicted, actual, risks)
```



EXAMPLE APPLICATIONS

- ATO Application: What life events affect compliance?
 - First application of the technology — 1995
 - Decision Stumps: $\text{Age} > \text{NN}$; Change in Marital Status
- Boosted Neural Networks
 - OCR using neural networks as base learners
 - Drucker, Schapire, Simard, 1993



SUMMARY

- ➊ Boosting is implemented in R in the `ada` library
- ➋ AdaBoost uses e^{-m} ; LogitBoost uses $\log(1 + e^{-m})$; Doom II uses $1 - \tanh(m)$
- ➌ AdaBoost tends to be sensitive to noise (addressed by BrownBoost)
- ➍ AdaBoost tends not to overfit, and as new models are added, generalisation error tends to improve.
- ➎ Can be proved to converge to a perfect model if the learners are always better than chance.

OVERVIEW

- 1 OVERVIEW
- 2 MULTIPLE MODELS
- 3 BOOSTING
 - Algorithm
 - Example
- 4 RANDOM FORESTS
 - Forests of Trees
 - Introduction
- 5 OTHER ENSEMBLES
 - Ensembles of Different Models



RANDOM FORESTS

- Original idea from Leo Breiman and Adele Cutler.
- The name is Licensed to Salford Systems!
- Hence, R package is randomForest.
- Typically presented in context of decision trees.
- Random Multinomial Logit uses multiple multinomial logit models.



RANDOM FORESTS

- Build many decision trees (e.g., 500).
- For each tree:
 - Select a random subset of the training set (N);
 - Choose different subsets of variables for each node of the decision tree ($m \ll M$);
 - Build the tree without pruning (i.e., overfit)
- Classify a new entity using every decision tree:
 - Each tree “votes” for the entity.
 - The decision with the largest number of votes wins!
 - The proportion of votes is the resulting score.

EXAMPLE: RF ON WEATHER DATA

```

set.seed(42)
(m <- randomForest(RainTomorrow ~ ., weather[train, -c(1:2, 23)],
  na.action=na.roughfix,
  importance=TRUE))

##
## Call:
## randomForest(formula=RainTomorrow ~ ., data=weath...
##           Type of random forest: classification
##           Number of trees: 500
## No. of variables tried at each split: 4
##
##           OOB estimate of  error rate: 13.67%
## Confusion matrix:
##           No Yes class.error
## No    211    4      0.0186
## Yes    31   10      0.7561

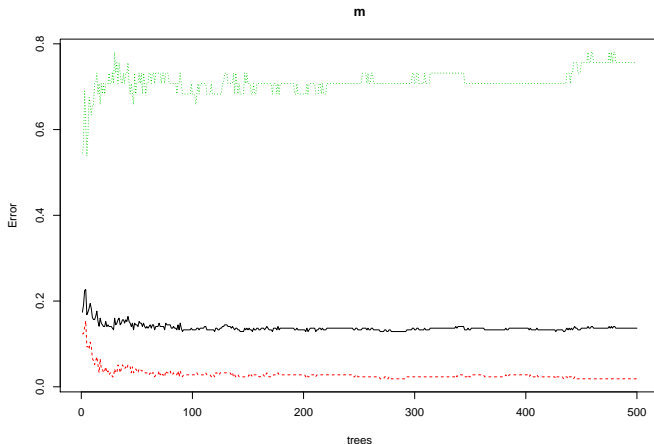
```



EXAMPLE: ERROR RATE

Error rate decreases quickly then flattens over the 500 trees.

`plot(m)`

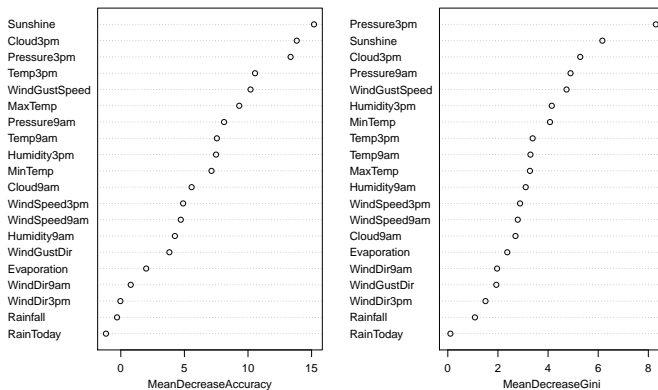


EXAMPLE: VARIABLE IMPORTANCE

Helps understand the *knowledge* captured.

```
varImpPlot(m, main="Variable Importance")
```

Variable Importance



EXAMPLE: SAMPLE TREES

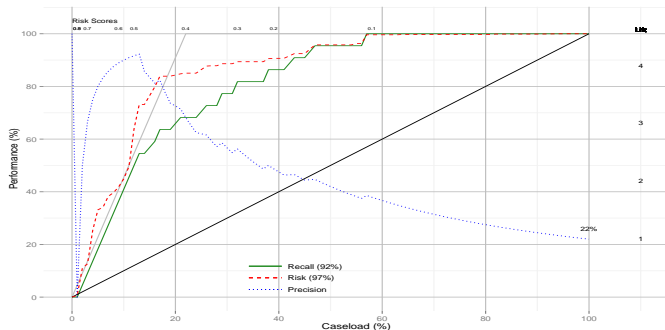
There are 500 trees in all. Here's some rules from the first tree.

```
## Random Forest Model 1
##
## -----...
## Tree 1 Rule 1 Node 30 Decision No
##
## 1: Evaporation <= 9
## 2: Humidity3pm <= 71
## 3: Cloud3pm <= 2.5
## 4: WindDir9am IN ("NNE")
## 5: Sunshine <= 10.25
## 6: Temp3pm <= 17.55
## -----...
## Tree 1 Rule 2 Node 31 Decision Yes
##
## 1: Evaporation <= 9
## 2: Humidity3pm <= 71
....
```



EXAMPLE: PERFORMANCE

```
predicted <- predict(m, weather[-train,], type="prob")[,2]
actual <- weather[-train,]$RainTomorrow
risks <- weather[-train,]$RISK_MM
riskchart(predicted, actual, risks)
```



FEATURES OF RANDOM FORESTS: BY BRIEMAN

- Most accurate of current algorithms.
- Runs efficiently on large data sets.
- Can handle thousands of input variables.
- Gives estimates of variable importance.



OVERVIEW

- 1 OVERVIEW
- 2 MULTIPLE MODELS
- 3 BOOSTING
 - Algorithm
 - Example
- 4 RANDOM FORESTS
 - Forests of Trees
 - Introduction
- 5 OTHER ENSEMBLES
 - Ensembles of Different Models

OTHER ENSEMBLES

- Netflix
 - Movie rental business - 100M customer movie ratings
 - \$1M for 10% improved root mean square error
 - First annual award (Dec '07) to KorBell (AT&T) 8.43% \$50K
 - Aggregate of the best other models!
 - Linear combination of 107 other models
 - <http://stat-computing.org/newsletter/v182.pdf>
- A lot of the different model builders deliver similar performance.
- So why not build one of each model and combine!
- In Rattle: Generate a Score file from all the models, and reload that into Rattle to explore.



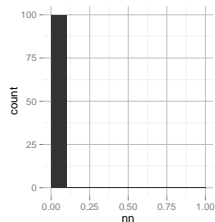
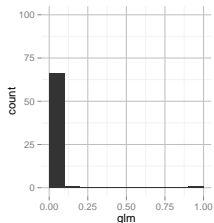
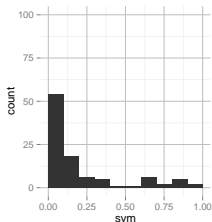
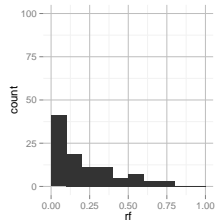
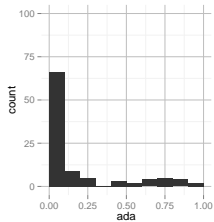
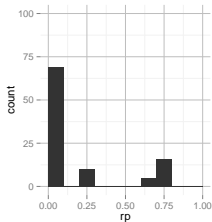
BUILD A MODEL OF EACH TYPE

```
ds <- weather[train, -c(1:2, 23)]
form <- RainTomorrow ~ .
m.rp  <- rpart(form, data=ds)
m.ada <- ada(form, data=ds)
m.rf  <- randomForest(form, data=ds, na.action=na.roughfix,
                       importance=TRUE)
m.svm <- ksvm(form, data=ds, kernel="rbfdot", prob.model=TRUE)
m.glm <- glm(form, data=ds, family=binomial(link="logit"))
m.nn  <- nnet(form, data=ds, size=10, skip=TRUE,
              MaxNWts=10000, trace=FALSE, maxit=100)
```

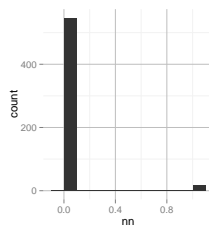
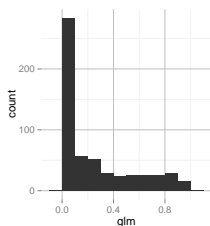
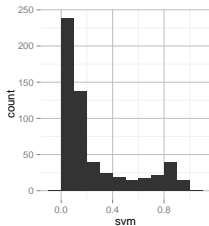
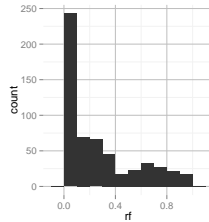
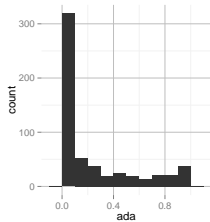
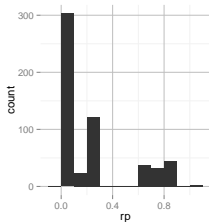
CALCULATE PROBABILITIES

```
ds <- weather[-train, -c(1:2, 23)]
ds <- na.omit(ds, "na.action")
pr <- data.frame(
  obs=row.names(ds),
  rp=predict(m.rp, ds)[,2],
  ada=predict(m.ada, ds, type="prob")[,2],
  rf=predict(m.rf, ds, type="prob")[,2],
  svm=predict(m.svm, ds, type="probabilities")[,2],
  glm=predict(m.glm, type="response", ds),
  nn=predict(m.nn, ds))
prw <- pr
```

PLOTS—WEATHER DATASET

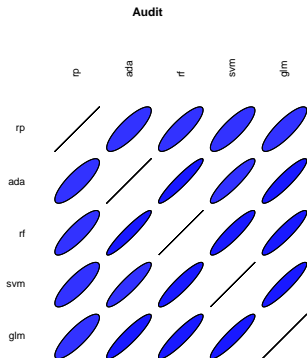
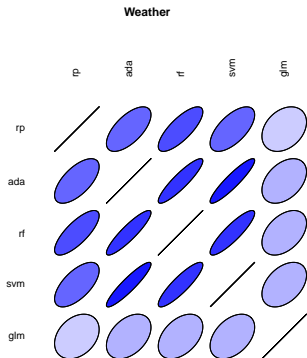


PLOTS—AUDIT DATASET



CORRELATION OF SCORES

The correlations between scores obtained by the different models suggest quite an overlap in their abilities to extract the same knowledge.

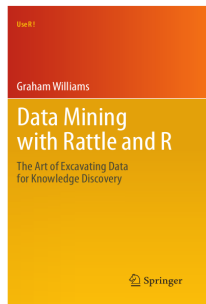


OVERVIEW

- 1 OVERVIEW
- 2 MULTIPLE MODELS
- 3 BOOSTING
 - Algorithm
 - Example
- 4 RANDOM FORESTS
 - Forests of Trees
 - Introduction
- 5 OTHER ENSEMBLES
 - Ensembles of Different Models



REFERENCE BOOK



Data Mining with Rattle and R

Graham Williams

2011, Springer, Use R!

ISBN: 978-1-4419-9889-7.

Chapters 12 and 13.



SUMMARY

- Ensemble: Multiple models working together
- Often better than a single model
- Variance and bias of the model are reduced
- The best available models today - accurate and robust
- In daily use in very many areas of application



SUMMARY

- Ensemble: Multiple models working together
- Often better than a single model
- Variance and bias of the model are reduced
- The best available models today - accurate and robust
- In daily use in very many areas of application

