CS 660: Mathematical Foundations for Analytics

Dr. Francis Parisi

Pace University

Spring 2017

Course Overview

1. Part I: Data Science Fundamentals
   - Data Science Concepts and Process
   - The R Language
   - Exploratory Data Analysis
   - Cleaning & Manipulating Data
   - Presenting Results
2. Part II: Graphs & Statistical Methods
   - Basic Graphics
   - Advanced Graphics
   - Probability & Statistical Methods
3. Part III: Modeling Methods
   - Model Selection and Evaluation
   - Linear and Logistic Regression
   - Unsupervised Methods
   - Advanced Modeling Methods

Classification Models

- Machine Learning Methods
- Demonstrate classification models
- Logistic Regression
- Decision Trees
- Random Forests
- Support Vector Machines
- Evaluating Models
- Data Mining with `rattle()`

Classification Models

- We develop classification models so we can predict future observations
- Classification methods predict to which class an observation belongs based on its features
- We start with EDA then partition the data
- For these examples we'll create a training data set and a validation set
- We build the model with the training data
- We evaluate how well the model does with the validation data

Classification Models - Setting up

```
pkgs <-  c("rpart", "rpart.plot", "parity", "randomForest", "e1071")
install.packages(pkgs, dependencies = TRUE)

loc <- "http://archive.ics.uci.edu/ml/machine-learning-databases/"
ds <- "breast-cancer-wisconsin/breast-cancer-wisconsin.data"
url <- paste(loc, ds, sep="")

breast <- read.table(url, sep=",", header=FALSE, na.strings="?")
names(breast) <- c("ID", "clumpThickness", "sizeUniformity",
   "shapeUniformity", "maginalAdhesion",
   "singleEpithelialCellSize", "bareNuclei",
   "blandChromatin", "normalNucleoli", "mitosis", "class")

df <- breast[-1]
df$class <- factor(df$class, levels=c(2,4),
   labels=c("benign", "malignant"))

set.seed(1234)
train <- sample(nrow(df), 0.7*nrow(df))
df.train <- df[train,]
df.validate <- df[-train,]
table(df.train$class)
table(df.validate$class)
```

Classification Models - Logistic Regression

- In our earlier discussion of logistic regression we learned that model predicts the logit, or log of the odds ratio
- We can exponentiate the logit to get the odds ratio
- ...and we can ”undo” the transformation to get probability of the event we are modeling
- If we set a threshold for the predicted probability we can turn the model into a classifier

Classification Models - Logistic Regression

```
fit.logit <- glm(class~., data=df.train, family=binomial())
summary(fit.logit)

prob <- predict(fit.logit, df.validate, type="response")
logit.pred <- factor(prob > .5, levels=c(FALSE, TRUE),
    labels=c("benign", "malignant"))
logit.perf <- table(df.validate$class, logit.pred,
    dnn=c("Actual", "Predicted"))
logit.perf
```

- We create a binary outcome from the probability by setting the threshold at 0.5
- Any observation with a predicted probability greater than 0.5 is considered malignant (in this example)
- Lastly, we create a confusion matrix

Classification Models - Decision Trees

- Decision trees are popular in data mining
- Starting at the top (root) we follow a set of binary splits that can be used to classify new observations
- Two types include classical trees and conditional trees
- A classical tree segregates the observations based on homogeneity
- A conditional tree segregates based on significance test

Classification Models - Decision Trees

1. Choose a predictor variable that splits the data into two groups and that maximizes homogeneity
2. Separate the data into the two groups and repeat the process for of the two new groups
3. Continue 1 and 2 until no splits reduce the impurity
4. Classify an observation by going down the tree until you reach terminal node

Classification Models - Decision Trees

```
library(rpart)
set.seed(1234)
dtree <- rpart(class ~ ., data=df.train, method="class",
               parms=list(split="information"))
dtree$cptable

plotcp(dtree)
dtree.pruned <- prune(dtree, cp=.0125)

library(rpart.plot)
prp(dtree.pruned, type = 2, extra = 104,
                  fallen.leaves = TRUE, main="Decision Tree")
dtree.pred <- predict(dtree.pruned, df.validate, type="class")
dtree.perf <- table(df.validate$class, dtree.pred,
                   dnn=c("Actual", "Predicted"))
dtree.perf
```

Classification Models - Random Forest

- A random forest is an ensemble learning approach
- Many predictive models are developed then aggregated
- If we have Nobservations in the training sample and M variables, then the algorithm is as follows:
  1. Grow a large number of decision trees by sampling N cases with replacement from the training set
  2. Sample $m < M$ variables at each node keep m constant at each node; these variables are considered candidates for splitting in that node
  3. Grow each tree fully without pruning (the minimum node size is set to 1)
  4. Terminal nodes are assigned to a class based on the mode of cases in that node
  5. Classify new observations by sending them down all the trees tracking the outcomesmajority rules
- Build random forests with `randomForest()` in the `random-Forest` package

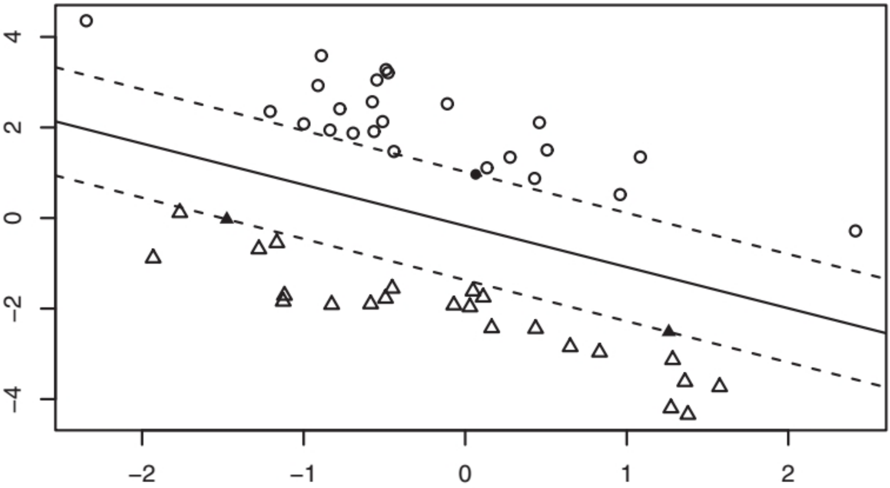
Classification Models - Random Forest

```
library(randomForest)
set.seed(1234)
fit.forest <- randomForest(class~., data=df.train,
               na.action=na.roughfix, importance=TRUE)

fit.forest

forest.pred <- predict(fit.forest, df.validate)
forest.perf <- table(df.validate$class, forest.pred,
                     dnn=c("Actual", "Predicted"))
forest.perf
```

Classification Models - Support Vector Machines

- A group of supervised machine-learning methods
- Can be used for classification and regression
- We seek an optimal *hyperplane* for separating two classes in a multidimensional space
- The chosen hyperplane maximizes the margin between the two classes closest points
- The points on the boundary of the margin are called support vectors (they help define the margin)
- The middle of the margin is the separating hyperplane
- For an N-dimensional space the optimal hyperplane has $N - 1$ dimensions
  - If there are two variables, the surface is a line
  - For three variables, the surface is a plane
  - For 10 variables, the surface is a 9-dimensional hyperplane

Classification Models - Support Vector Machines



Classification Models - Support Vector Machines

```
library(e1071)
set.seed(1234)
fit.svm <- svm(class~., data=df.train)
fit.svm

svm.pred <- predict(fit.svm, na.omit(df.validate))
svm.perf <- table(na.omit(df.validate)$class,
                  svm.pred, dnn=c("Actual", "Predicted"))
svm.perf
```

Classification Models - Choosing the Best Model

- We can use the measures we discussed previously: accuracy, sensitivity, specificity to evaluate how well our model works
- Instead of calculating each separately we can borrow a function defined in the book *R in Action*

Classification Models - Choosing the Best Model

```
performance <- function(table, n=2){
if(!all(dim(table) == c(2,2)))
            stop("Must be a 2 x 2 table")
tn = table[1,1]
fp = table[1,2]
fn = table[2,1]
tp = table[2,2]
sensitivity = tp/(tp+fn)
specificity = tn/(tn+fp)
ppp = tp/(tp+fp)
npp = tn/(tn+fn)
hitrate = (tp+tn)/(tp+tn+fp+fn)
result <- paste("Sensitivity = ", round(sensitivity, n),
     "\nSpecificity = ", round(specificity, n),
     "\nPositive Predictive Value = ", round(ppp, n),
     "\nNegative Predictive Value = ", round(npp, n),
     "\nAccuracy = ", round(hitrate, n), "\n", sep="")
     cat(result)
}
```

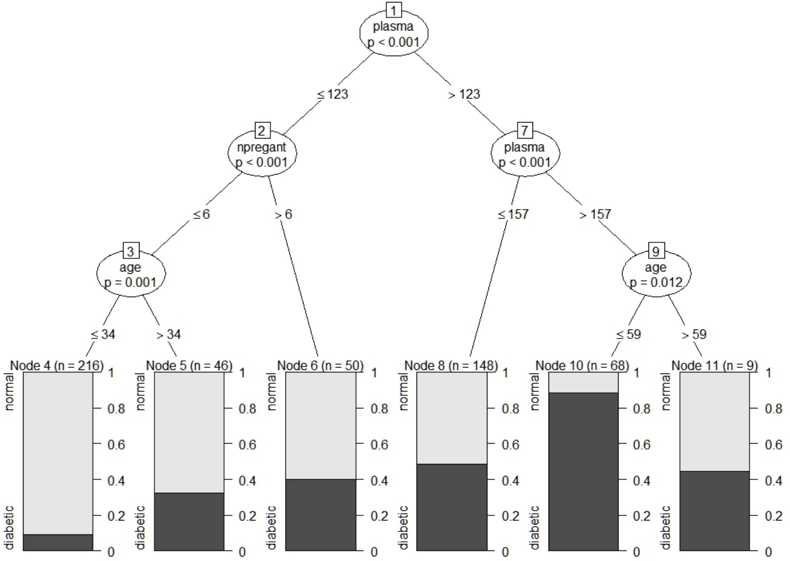
Classification Models - Data Mining with Rattle

- Rattle – **R** **A**nalytic **T**ool **T**o **L**earn **E**asily
- GUI for data mining in R
- Point-and-click access to supervised and unsupervised models
- Includes the ability to transform data and has data-visualization tools
- `install.packages("rattle")`
- `library(rattle)`
  `rattle()` Launches the rattle interface

See Williams (2011) for more on Rattle

Classification Models - Data Mining with Rattle

```
loc <- "http://archive.ics.uci.edu/ml/machine-learning-databases/"
ds <- "pima-indians-diabetes/pima-indians-diabetes.data"
url <- paste(loc, ds, sep="")
diabetes <- read.table(url, sep=",", header=FALSE)

names(diabetes) <- c("npregant", "plasma", "bp", "triceps",
              "insulin", "bmi", "pedigree", "age", "class")
diabetes$class <- factor(diabetes$class, levels=c(0,1),
labels=c("normal", "diabetic"))
library(rattle)
rattle()

cv <- matrix(c(145, 50, 8, 27), nrow=2)
performance(as.table(cv))
```



1
plasma
p < 0.001
≤ 123
> 123
2
npregant
p < 0.001
7
plasma
p < 0.001
≤ 6
> 6
≤ 157
> 157
3
age
p = 0.001
9
age
p = 0.012
≤ 34
> 34
≤ 59
> 59
Node 4 (n = 216)
Node 5 (n = 46)
Node 6 (n = 50)
Node 8 (n = 148)
Node 10 (n = 68)
Node 11 (n = 9)
normal
diabetic
1
0.8
0.6
0.4
0.2
0

Classification Models – Summary

- We looked at several machine-learning methods for classifying observations into one of two groups
- The methods vary from low complexity like logistic regression and decision trees to high complexity like random forests and support vector machines
- Classification models apply to many fields (beyond medicine): computer science, finance, marketing, etc.
- We looked at problems with two grouops but these methods extend to multigroup classification problems

References

James, G., Hastie, T., Witten, D. and Tibshirani, R. (2013).
An Introduction to Statistical Learning with Applications in R.
Springer, second edition.
6th Printing 2015.

Kabacoff, R. I. (2015).
R in Action.
Manning, Shelter Island, NY, second edition.

Lander, J. P. (2014).
R for Everyone.
Addison-Wesley, Upper Saddle River.

Williams, G. (2011).
Data Mining with Rattle and R.
Springer, New York.

Zumel, N. and Mount, J. (2014).
Practical Data Science with R.
Manning, Shelter Island, NY, second edition.