Predictive Analytics Machine Learning Lab Handout

Data Matters, Odum Institute, UNC/Chapel Hill, 26 June 2014
Instructor: Philip Schrodt, Parus Analytical Systems [schrodt735@gmail.com]

References

Read the data and create random train/test sets

Get into the appropriate directory, then

```
afrdata <- read.table("PredAnalytics.Africa.data.csv", header=TRUE,  sep=",")

# make sure it looks okay
names(afrdata)
afrdata[1:16,]

# some methods don't allow NA, so just get complete cases
allafr = afrdata[complete.cases(afrdata),]
```

Generate a binary variable for DEMOC

```
allafr$bidemoc <- ifelse(allafr$DEMOC >= 5, 1,0)
table(allafr$bidemoc)
```

Generate a 2:1 training and test set

```
index <- 1:nrow(allafr)
testindex <- sample(index, trunc(length(index)/3))
testset <- allafr[testindex,]
trainset <- allafr[-testindex,]
```

SVM

```
library(e1071)
svm.model <- svm(bidemoc ~ ., data = trainset, cost = 100, gamma = 1)
x <- subset(trainset, select = -bidemoc)
y <- trainset$bidemoc
pred <- predict(svm.model, x)
plot(pred,y)
xt <- subset(testset, select = -bidemoc)
predt <- predict(svm.model, xt)
plot(predt,testset$bidemoc)
table(pred = predt>0.5,true =testset$bidemoc)
```

Listing false positives

```
dat3 = data.frame(predt,testset$bidemoc,testset$YEAR,testset$NAME)
fnset = subset(dat3,predt<.5 & testset.bidemoc == 1,
            select = c("testset.NAME","testset.YEAR"))
fnset
```

Metrics with ROCR

http://rocr.bioinf.mpi-sb.mpg.de/

http://rocr.bioinf.mpi-sb.mpg.de/ROCR.pdf

ROCR has literally 35 different measures, mostly functions of the cutoff. This provides numerous opportunities for elaborate graphs; I'm just showing the simple one.

```
library(ROCR)
rocrpred <- prediction(predt,testset$bidemoc)
perf <- performance(rocrpred, measure = "tpr", x.measure = "fpr")
plot(perf)
aucval <- performance(rocrpred, measure = "auc")
as.numeric(aucval@y.values)
perf <- performance(rocrpred, measure = "acc", x.measure = "cutoff")
plot(perf, col=rainbow(10))  # add some color
```

Neural network

http://www.r-bloggers.com/using-neural-networks-for-credit-scoring-a-simple-example/

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

Geoffrey Hinton's course: https://www.coursera.org/course/neuralnets

```
library(neuralnet)
thenet <- neuralnet(bidemoc ~ GDPCAP + REGTYPE + IMR, trainset,
+ hidden = 8, lifesign = "minimal", linear.output = FALSE, threshold = 0.1)
temp_test = subset(testset,select = c("GDPCAP","REGTYPE","IMR"))
prednn = compute(thenet,temp_test)
plot(prednn$net.result,testset$bidemoc)
znn = ifelse(prednn$net.result >= .5, 1,0)
table(pred = znn, true = testset$bidemoc)
```

Update 26 June: In the exercise in class, we got extremely erratic results with this—it did not consistently converge. So this may not be the best of examples.

Random forest

http://rinparis.blogspot.com/2011/10/learn-random-forest-by-example-iris.html

```
library(randomForest)
indvar <- c("RIOTS","ILLITFM","ETHVIOL","IMR")
therf <- randomForest(bidemoc~RIOTS + ILLITFM+ ETHVIOL  + IMR , data=trainset ,
             ntree=50, proximity=TRUE)

inddat <- trainset[indvar]
predr <- predict(therf, inddat)
plot(predr,trainset$bidemoc)
zr <- ifelse(predr >= .5, 1,0)
table(zr,trainset$bidemoc)

testdat <- testset[indvar]
predtr <- predict(therf, testdat)
ztr <- ifelse(predtr >= .5, 1,0)
table(ztr,testset$bidemoc)
```

This model ends up with a large number of false negatives in both the training and the test sets, probably because of the imbalance towards negative cases. That might be improved by using all of the positive cases but a smaller fraction of the negative.

Hierarchical clustering

```
library(cluster)
somevars = c("YEAR","NAME","GDPCAP", "ILLITFM", "IMR", "IMPORTS", "ETHVIOL")
scaleafr = allafr[somevars]
scaleafr[,3:7] <- scale(scaleafr[,3:7])  # just scale the values with are continuous mea
scaleafr[1:8,]
dat2 = subset(scaleafr,YEAR == 1995)  # take only one year so we can actually see the la
d <- dist(dat2[,3:7], method = "euclidean")
fit <- hclust(d, method="ward")
plot(fit, labels = dat2$NAME)
```