R Programs for Text Book Examples

Chapter 11: Neural Nets

Code for neural network with a single hidden layer (three nodes) for the data example on tasting scores for six consumers and two predictors (Table 11.1 and Figure 11.2)

####

```
library(neuralnet)
df <- read.csv("Tinydata.csv")

df$Like <- df$Acceptance=="like"
df$Dislike <- df$Acceptance=="dislike"

set.seed(1)
nn <- neuralnet(Like + Dislike ~ Salt + Fat, data = df, linear.output = F, hidden = 3)

# display weights
nn$weights

# display predictions
prediction(nn)

# plot network
plot(nn, rep="best")
```

Code for generating confusion matrix for the data example (Figure 11.4)

####

```
library(caret)
predict <- compute(nn, data.frame(df$Salt, df$Fat))
predicted.class=apply(predict$net.result,1,which.max)-1
confusionMatrix(ifelse(predicted.class=="1", "dislike", "like"), df$Acceptance)
```

Code for a neural network with two nodes in the hidden layer (accidents data) and output for neural network for accident data with two nodes in the hidden layer (Table 11.7, 11.8)

####

```
library(neuralnet,nnet,caret)

accidents.df <- read.csv("Accidents.csv")
# selected variables
vars <- c("ALCHL_I", "PROFIL_I_R", "VEH_INVL")

# partition the data
set.seed(2)
```

```
training=sample(row.names(accidents.df), dim(accidents.df)[1]*0.6)
validation=setdiff(row.names(accidents.df), training)

# when y has multiple classes - need to dummify
trainData <- cbind(accidents.df[training,c(vars)],
            class.ind(accidents.df[training,]$SUR_COND),
            class.ind(accidents.df[training,]$MAX_SEV_IR))
names(trainData) <- c(vars,
            paste("SUR_COND_", c(1, 2, 3, 4, 9), sep=""), paste("MAX_SEV_IR_", c(0, 1, 2), sep=""))

validData <- cbind(accidents.df[validation,c(vars)],
            class.ind(accidents.df[validation,]$SUR_COND),
            class.ind(accidents.df[validation,]$MAX_SEV_IR))
names(validData) <- c(vars,
            paste("SUR_COND_", c(1, 2, 3, 4, 9), sep=""), paste("MAX_SEV_IR_", c(0, 1, 2), sep=""))

# run nn with 2 hidden nodes
# use hidden= with a vector of integers specifying number of hidden nodes in each layer
nn <- neuralnet(MAX_SEV_IR_0 + MAX_SEV_IR_1 + MAX_SEV_IR_2 ~
            ALCHL_I + PROFIL_I_R + VEH_INVL + SUR_COND_1 + SUR_COND_2
           + SUR_COND_3 + SUR_COND_4, data = trainData, hidden = 2)

training.prediction <- compute(nn, trainData[,-c(8:11)])
training.class <- apply(training.prediction$net.result,1,which.max)-1
confusionMatrix(training.class, accidents.df[training,]$MAX_SEV_IR)

validation.prediction <- compute(nn, validData[,-c(8:11)])
validation.class <-apply(validation.prediction$net.result,1,which.max)-1
confusionMatrix(validation.class, accidents.df[validation,]$MAX_SEV_IR)
```

Chapter 12: Discriminant Analysis

Code for discriminant analysis for riding-mower data, displaying the estimated classification functions (Figure 12.3)

```
####
library(DiscriMiner)
mowers.df <- read.csv("RidingMowers.csv")
da.reg <- linDA(mowers.df[,1:2], mowers.df[,3])
da.reg$functions
```

Code for classification scores, predicted classes, and probabilities for riding-mower data (Figure 12.4)

```
####
da.reg <- linDA(mowers.df[,1:2], mowers.df[,3])
# compute probabilities manually (below); or, use lda() in package MASS with predict()
propensity.owner <- exp(da.reg$scores[,2])/(exp(da.reg$scores[,1])+exp(da.reg$scores[,2]))
```

```
data.frame(Actual=mowers.df$Ownership,
        da.reg$classification, da.reg$scores, propensity.owner=propensity.owner)
```

Code for discriminant analysis for the three-class injury example: classification functions and confusion matrix for training set (Figure 12.8)

```
####
library(DiscriMiner)
library(caret)

accidents.df <-  read.csv("Accidents.csv")
lda.reg <- linDA(accidents.df[,1:10], accidents.df[,11])
lda.reg$functions
confusionMatrix(da.reg$classification, accidents.df$MAX_SEV)
```

Code for classification scores, membership probabilities, and classifications for the three-class injury training dataset (Figure 12.9)

```
####
propensity <- exp(lda.reg$scores[,1:3])/
  (exp(lda.reg$scores[,1])+exp(lda.reg$scores[,2])+exp(lda.reg$scores[,3]))

res <- data.frame(Classification = lda.reg$classification,
              Actual = accidents.df$MAX_SEV,
              Score = round(da.reg$scores,2),
              Propensity = round(propensity,2))
head(res)
```