Chapter 7 – K-Nearest-Neighbor

Instructor: Zach Zhizhong ZHOU,
Shanghai Jiao Tong University
主讲教师 :周志中，上海交通大学

Data Mining for Business Intelligence

Shmueli, Patel & Bruce

Riding Mowers - 1

```
library(class) ## needed for KNN
setwd("C:/BA/KNN")
RidingMowers <- read.csv("RidingMowers.csv",header =
TRUE)
RM <- cbind.data.frame ( scale(RidingMowers$Income),
            scale(RidingMowers$Lot_Size))
#上面的语句对Income和LotSize进行标准化，然后把2列重新组
合成一个dataframe（使用cbind.data.frame语句）
#如果使用cbind语句，那么factor型数据会被强制转化成数值
（level值），而不是factor型数据的值。
#所以应该使用cbind.data.frame语句
#可以试一下 cbind (RidingMowers$Ownership) 看有什么结果。
colnames(RM) <- c("Income","LotSize") #对2个列重命名。
```

Riding Mowers - 2

```
#训练数据集有18个数据，我们打算对剩下6个数据进行分类。
set.seed(400)
#试一下set.seed(200)看有什么结果？设置随机数种子是为了重现结果。
trainidx <- sample(1:24,18)
mytrain <- RM[trainidx,]
#mytrain是训练数据集的预测变量
mytest <- RM[-trainidx,]     #mytest是验证数据集的预测变量
classification_of_train <- RidingMowers[trainidx,"Ownership"]
#classification_of_train是训练数据集的结果变量
classification_of_test <- RidingMowers[-trainidx,"Ownership"]
#classification_of_test是验证数据集的结果变量
```

Riding Mowers - 3

```
nearest1 <- knn(train = mytrain, test = mytest,
classification_of_train, k=3)

nearest2 <- knn(train = mytrain, test = mytest,
classification_of_train, k=6)

data.frame(classification_of_test, nearest1, nearest2)

pcorrn1=sum(classification_of_test==nearest1)/6

pcorrn1

pcorrn2=sum(classification_of_test==nearest2)/6

pcorrn2
```

Riding Mowers 小结1

通过这个R Script，我们学到：

1. 如何把几列数据合并成一个数据框（cbind.data.frame函数）。
2. 如何对数据框的几列重命名（colnames(RM) <- c("Income","LotSize") ）
3. 如何使用knn函数。
4. 计算两个向量当中相等变量的个数
   （sum(classification_of_test==nearest1)）

Riding Mowers - 4

```
library(caret)

setwd("C:/BA/KNN")


RidingMowers <- read.csv("RidingMowers.csv",header = TRUE)


Predictors <- RidingMowers[,c("Income","Lot_Size")]
```

Riding Mowers - 5

model <- train(　Predictors, RidingMowers[,"Ownership"], method='knn',

#上面先列出Predictors, 然后是结果变量，然后说明使用KNN方法。

tuneGrid = data.frame(k=1:17),　#KNN模型当中k的取值范围从1到17。

metric='Accuracy',　#评价指标是“准确率” Accuracy

trControl=trainControl(method='repeatedcv',　number=4, repeats=20))

#trControl 是对训练过程进行控制的函数。此处的method='repeatedcv' 意思是使用repeated cross validation 方法（重复交叉验证）。number=4表示做4-fold cross validation，意思是把数据集割成4块，然后做4次训练和验证，每次都取其中一块数据（1/4的数据）当验证数据集，剩下的当训练数据集。repeat=30表示上面的过程重复20次，等总共要做80次训练-验证。最终计算评价指标（此处是Accuracy）的平均值。

Riding Mowers - 6

```
model

plot(model)

confusionMatrix(model)


Predictors_Example <- data.frame(Income=80, Lot_Size=20)

predict(model$finalModel,Predictors_Example)
```

Kappa检验（一致性检验）

看预测结果和实际结果吻合程度

Kappa = 1 预测结果与实际结果完全吻合。

Kappa≥0.75 预测结果与实际结果一致性较好。

0.75>Kappa≥0.4 预测结果与实际结果一致性一般。

Kappa<0.4 预测结果与实际结果一致性较差。

Riding Mowers 小结2

通过这个R Script，我们学到：

1. 如何使用train函数对数据集进行训练，找到KNN方法的最佳k值。
2. 如何展示混淆矩阵（confusion matrix）。
3. 如何使用训练得到的KNN模型预测新的记录目标值。