Data_Modeling_hw1

Po-Han, Lai

2019/8/22

R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see http://rmarkdown.rstudio.com.

When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

```
#install.packages("kernlab")
library(kernlab)
```

Including Plots

```
## [1] "data.frame"
```

```
##  Setting default kernel parameters
```

```
## [1] 0.08158492
```

```
## [1] 0.8639144
```

```
##  Setting default kernel parameters
## [1] 0.5474006
##  Setting default kernel parameters
## [1] 0.8639144
##  Setting default kernel parameters
## [1] 0.8639144
##  Setting default kernel parameters
## [1] 0.8639144
##  Setting default kernel parameters
## [1] 0.8623853
##  Setting default kernel parameters
## [1] 0.8639144
```

Use Different Kernel Rbfdot, polydot, anovadot

```
cost<-c()
accuracy<-c()
diff_model<-c()
kernel_choice<-c("rbfdot","polydot","vanilladot","anovadot")
for (i in c){
  for (w in kernel_choice){
  model <- ksvm(header[,1:10],header[,11],type="C-svc",kernel=w,C=i,scaled=TRUE)
  pred=predict(model,header[,1:10])
  cost<-append(cost,i)
  acc<-sum(pred==header[,11])/nrow(header)
  accuracy<-append(accuracy,acc)
  diff_model<-append(diff_model,w)}
  }
```

```
##  Setting default kernel parameters
##  Setting default kernel parameters
##  Setting default kernel parameters
##  Setting default kernel parameters
##  Setting default kernel parameters
##  Setting default kernel parameters
##  Setting default kernel parameters
##  Setting default kernel parameters
##  Setting default kernel parameters
##  Setting default kernel parameters
##  Setting default kernel parameters
##  Setting default kernel parameters
##  Setting default kernel parameters
##  Setting default kernel parameters
##  Setting default kernel parameters
##  Setting default kernel parameters
##  Setting default kernel parameters
```

```
comparison<-data.frame(cost=cost,accuracy=accuracy,model=diff_model)

comparison<-comparison[order(-accuracy),]
head(comparison,5)
```

```
##     cost  accuracy    model
## 21 1e+05 0.9969419   rbfdot
## 17 1e+04 0.9908257   rbfdot
## 13 1e+02 0.9510703   rbfdot
## 20 1e+04 0.9082569 anovadot
## 16 1e+02 0.9067278 anovadot
```

```
best_model <- ksvm(header[,1:10],header[,11],type="C-svc",kernel="rbfdot",C=10000,scaled=TRUE)
pred=predict(best_model,header[,1:10])

table(pred,header[,11])
```

```
##
## pred   0   1
##    0 358   3
##    1   0 293
```

Note that the `echo = FALSE` parameter was added to the code chunk to prevent printing of the R code that generated the plot.

```
#install.packages("kknn")
library(kknn)
header.df<-as.data.frame(header)

knn_accuracy = function(k){
  prediction<-c()
  for (i in 1:nrow(header.df)){
    k_model<-kknn(R1~.,header.df[-i,],header.df[i,],k=k,scale=TRUE)
    prediction_ans<-ifelse(predict(k_model)>=0.5,1,0)
    prediction<-append(prediction,prediction_ans)}
    accuracy=sum(prediction==header.df[,11])/nrow(header.df)
    return(accuracy)
```

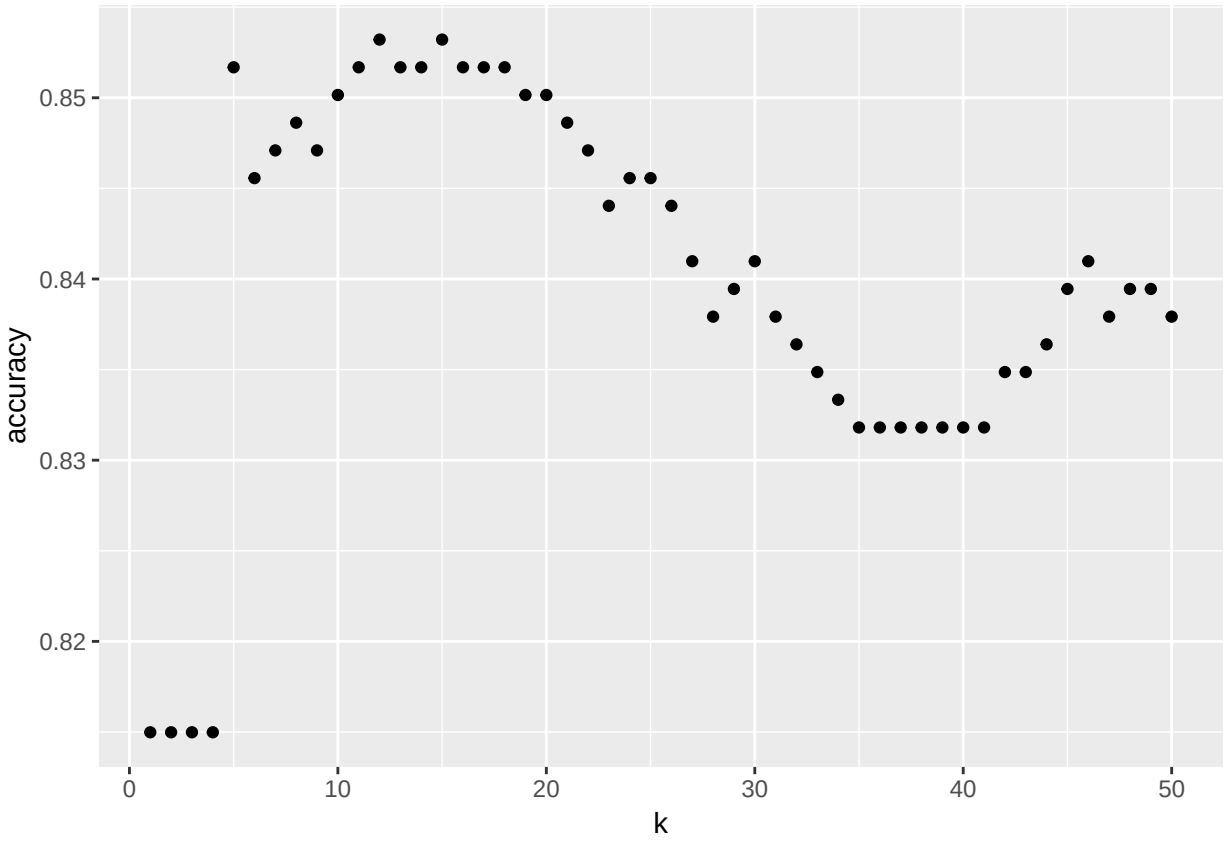
```
}

which_k=c()
for (i in 1:50){
  which_k[i]<-knn_accuracy(i)
}

k_model<-data.frame(accuracy<-which_k,k=1:50)
```

```
library(ggplot2)

ggplot(aes(x=k,y=accuracy),data=k_model)+geom_point()
```




```
ratio=round(nrow(header.df)*0.2)

sample.index<-sample(1:nrow(header.df),size=ratio,replace=FALSE)

train<-header.df[-sample.index,]
test<-header.df[sample.index,]

model2<-train.kknn(R1~.,train, kmax = 100,kernel=c("optimal","rectangular","inv","gaussian","triangular

print(model2)
```

```
##
## Call:
## train.kknn(formula = R1 ~ ., data = train, kmax = 100, kernel = c("optimal",     "rectangular", "inv
```

```
##
## Type of response variable: continuous
## minimal mean absolute error: 0.2066768
## Minimal mean squared error: 0.1111811
## Best kernel: inv
## Best k: 35
```

```
compare_5_kernel<-as.data.frame(model2$MEAN.SQU)
```

```
library(tidyverse)
compare_5_kernel$k<-seq(1,100,1)
compare_5_kernel_viz<-compare_5_kernel %>% gather(key=kernel,value = mean_error,1:5)

ggplot(aes(x=k,y=mean_error,color=kernel),data=compare_5_kernel_viz) +geom_point()
```