Lab3_Block1_A14

Machine Learning – 732A99

Ahmet Hakan Akdeve(ahmak554), Zhixuan Duan(zhidu838), Jun li(junli559)

2019-12-14

Assignment 1: KERNEL METHODS

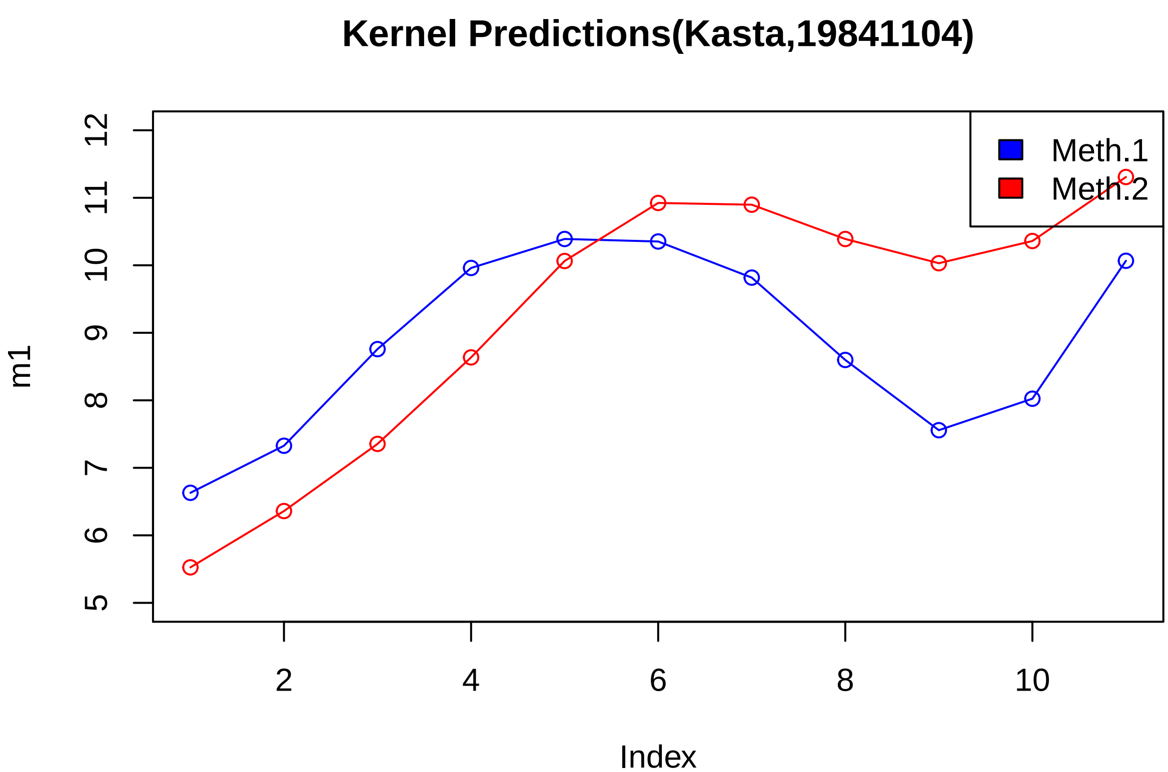
Part 1

The kernels' width are selected as reciprocal of variance of the training data in central-distance, numeric transformation of date and time, and Gaussian is adopted as kernel function, therefore when the observation is closer the exponential value will approach to 1, and vice versa.

The two kernel methods have different constructures, first one has sum of kernels while the other has a product as input.

```
##
## Attaching package: 'lubridate'

## The following object is masked from 'package:base':
##
##     date
```



Assignment 2: SUPPORT VECTOR MACHINES

```
## Source: local data frame [3 x 6]
## Groups: <by row>
##
## # A tibble: 3 x 6
##      TP    TN    FN    FP accuracy miscl_rate
##   <int> <int> <int> <int>    <dbl>      <dbl>
## 1  1244  2041   114    51    0.952     0.0478
## 2  1262  2049    96    43    0.960     0.0403
## 3  1306  2071    52    21    0.979     0.0212
```

The table above shows the result of how the models predict on training data. Model with C=5 seems to be the best one since it has the lowest misclassification rate. The misclassification rates are very low which can be an indication of overfitted models.

```
## Source: local data frame [3 x 6]
## Groups: <by row>
##
## # A tibble: 3 x 6
##      TP    TN    FN    FP accuracy miscl_rate
##   <int> <int> <int> <int>    <dbl>      <dbl>
## 1   381   669    74    27    0.912     0.0877
## 2   399   665    56    31    0.924     0.0756
## 3   397   664    58    32    0.922     0.0782
```

For test data, the models with c=3 and c=5 appear to be the best models since they have same lowest misclassification rate.

```r
svm_3 <- ksvm(x = type ~ ., data = train,
              kernel = rbfdot(sigma = 0.05),
              C = 5)
svm_3
```

```
## Support Vector Machine object of class "ksvm"
##
## SV type: C-svc  (classification)
##  parameter : cost C = 5
##
## Gaussian Radial Basis kernel function.
##  Hyperparameter : sigma =  0.05
##
## Number of Support Vectors : 1284
##
## Objective Function Value : -1577.025
## Training error : 0.021159
```

We choose to display the R-output for the model when c=5. The result is shown above.

Code Appendix

```r
## ----eval=TRUE,echo=FALSE,warning=FALSE,message=FALSE--------------------
RNGversion('3.5.1')


## ----eval=TRUE,echo=FALSE,warning=FALSE----------------------------------
## Assignment 1: KERNEL METHODS
## part 1
set.seed(1234567890)
library(geosphere)
library(kernlab)
library(lubridate)
stations <- read.csv("stations.csv")
temps <- read.csv("temps50k.csv")
st <- merge(stations,temps,by="station_number")

## preparing test data
a <- 58.4274 # The point to predict (up to the students)
b <- 14.826
date <- "1984-11-04" # The date to predict (up to the students)
times <- seq(4,24,2);len<-length(times)
for(i in 1:len) times[i]<-paste(times[i],":00:00",sep="")
times<-as.numeric(hms(times))
temp <- vector(length=len)

## prapare train data
da<-cbind(latitude=st$latitude,longitude=st$longitude,
          date=as.numeric(as.Date(st$date,origin="1900-01-01")),
          time=as.numeric(hms(st$time)),air_temperature=st$air_temperature)
train<-da[which(da[,3]<as.numeric(as.Date(date,origin="1900-01-01"))),];num<-nrow(train)
cen<-c(mean(train[,2]),mean(train[,1]))  ## mean point of coordinates

## kernel function and matrix
train_distance<-vector(length=num)
for(i in 1:num)
  train_distance[i]<-distHaversine(c(train[i,2],train[i,1]),cen)

k1<-rbfdot(sigma=1/var(train_distance))
k2<-rbfdot(sigma=1/var(train[,3]))
k3<-rbfdot(sigma=1/var(train[,4]))

h_distance<-kernelMatrix(k1,train_distance)
h_date <-kernelMatrix(k2,train[,3])
h_time <-kernelMatrix(k3,train[,4])

## method 1
km<-h_distance+h_date+h_time
gene1 <- ksvm(km,train[,5],kernel="matrix",C=10)

x1<-rep(distHaversine(c(b,a),cen),len)
x2<-rep(as.numeric(as.Date(date,origin="1900-01-01")),len)
#x<-cbind(x1,x2,times)
xkm1<-kernelMatrix(k1,x1,train_distance)
xkm2<-kernelMatrix(k2,x2,train[,3])
```

```r
xkm3<-kernelMatrix(k3,times,train[,4])
xkm<-as.kernelMatrix((xkm1+xkm2+xkm3)[,alphaindex(gene1)])
m1<-predict(gene1,xkm, type="response")

## method 2
km<-h_distance*h_date*h_time
gene2 <- ksvm(km,train[,5],kernel="matrix",C=10)
xkm<-as.kernelMatrix((xkm1*xkm2*xkm3)[,alphaindex(gene2)])
m2<-predict(gene2,xkm, type="response")

plot(m1, type="o",main="Kernel Predictions(Kasta,19841104)",col="blue",ylim=c(5,12))
lines(m2, type="o",col="red")
legend('topright',legend=c("Meth.1","Meth.2"),fill=c("blue","red"))


## ----include=FALSE-------------------------------------------------------

knitr::opts_chunk$set(echo = TRUE)
library(lubridate)
library(geosphere)
library(stringr)
library(kernlab)
library(xtable)
library(geosphere)
library(ggplot2)
library(readr)
library(dplyr)
library(pamr)
knitr::opts_chunk$set(fig.width=8, fig.height=5)
#setwd("C:\\Users\\Suat\\Desktop\\Master_courses\\732A99_MachineLearning\\Lab3")


## ----echo=FALSE----------------------------------------------------------

data(spam)

n <- dim(spam)[1]
set.seed(12345)
id <- sample(1:n, floor(n*0.75))
train <- spam[id, ]
test <- spam[-id, ]


set.seed(12345)
svm_1 <- ksvm(x = type ~ ., data = train,
                 kernel = rbfdot(sigma = 0.05),
                 C = 0.5)
set.seed(12345)
svm_2 <- ksvm(x = type ~ ., data = train,
                 kernel = rbfdot(sigma = 0.05),
                 C = 1)

set.seed(12345)
```

```r
svm_3 <- ksvm(x = type ~ ., data = train,
                 kernel = rbfdot(sigma = 0.05),
                 C = 5)

mods <- tibble(model = list(svm_1, svm_2, svm_3),train = train %>% list(),
                 test = test %>% list())


mods<- mods %>% rowwise() %>%
  mutate(preds_train = predict(object = model, newdata = train) %>% list(),
         preds_test = predict(object = model, newdata = test) %>% list(),
         confusion_mat = (table(preds_train, train %>% select(type) %>% unlist())
                          %>% list()),
         TP = confusion_mat[2, 2],TN = confusion_mat[1, 1],FN = confusion_mat[1, 2],
         FP = confusion_mat[2, 1],
         accuracy = (TP+TN)/(TP+TN+FP+FN),
         miscl_rate = 1 - accuracy)

mods[,7:12]



## ----echo=FALSE----------------------------------------------------------
mods <- mods %>%
  rowwise() %>%
  mutate(confusion_mat = (table(preds_test, test %>% select(type) %>% unlist())
                          %>% list()),
         TP = confusion_mat[2, 2],TN = confusion_mat[1, 1],FN = confusion_mat[1, 2],
         FP = confusion_mat[2, 1],
         accuracy = (TP+TN)/(TP+TN+FP+FN),
         miscl_rate = 1 - accuracy)


mods[,7:12]



## ----echo=TRUE-----------------------------------------------------------

svm_3 <- ksvm(x = type ~ ., data = train,
              kernel = rbfdot(sigma = 0.05),
              C = 5)
svm_3


## ----code = readLines(knitr::purl("C:/Users/A550240/Desktop/LIU/1.2_MachineLearning/Lab3_block1_5/Lab
## NA
```