Lab3 Block1

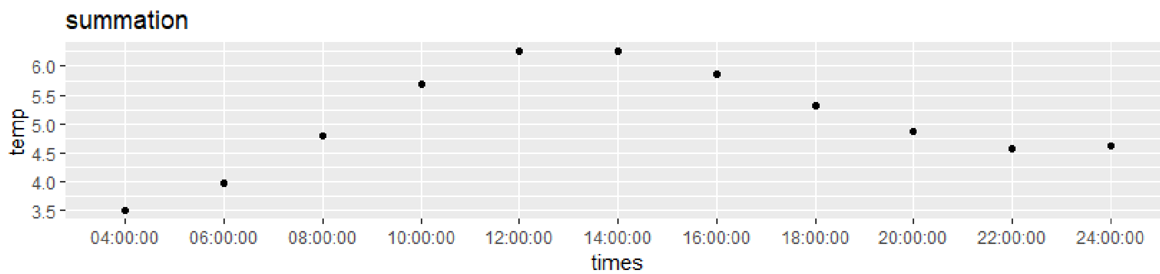
Group A5

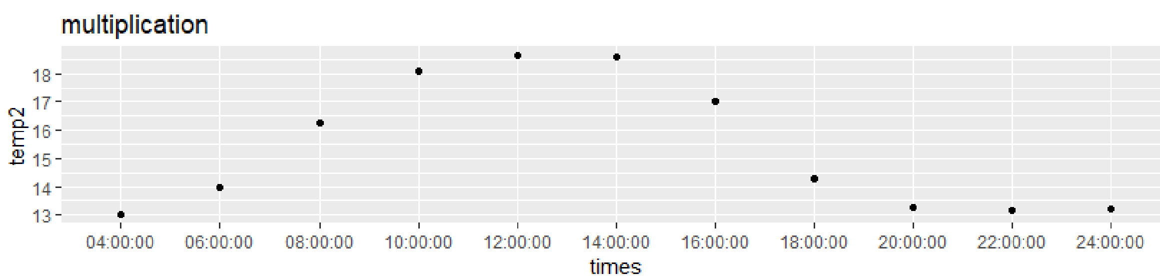
2018-12-18

Assignment 1

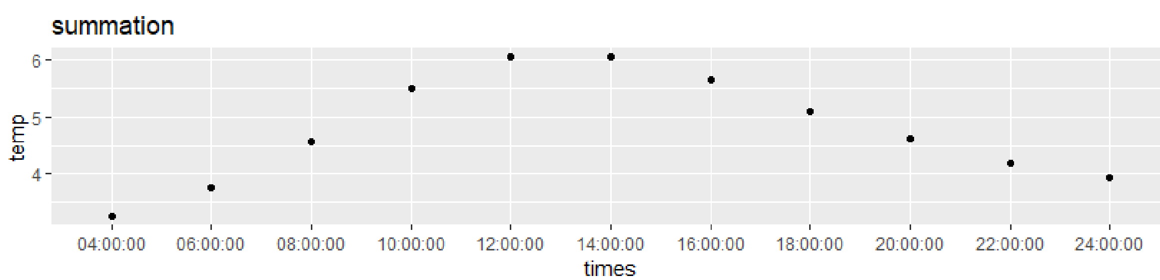
We implement a kernel method to predict the hourly temperatures for a date and place in Sweden. The h values are 33000, 40, 4 for distance, date and hour respectively. The following plots are the temperature predictions in Linkoping (58.4137,15.6235).

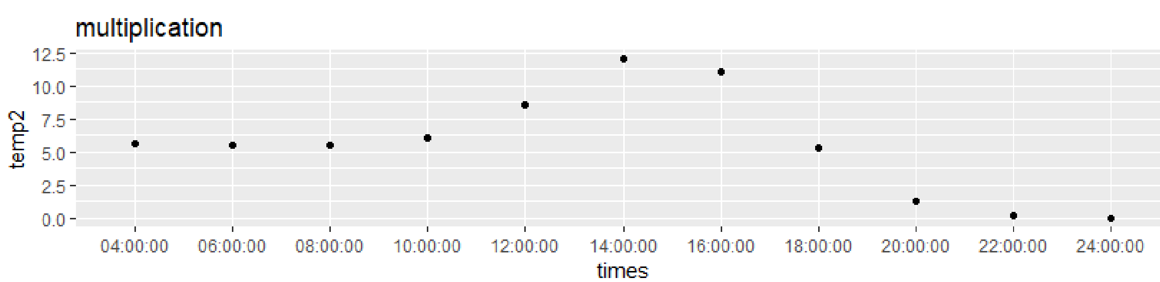
2018-7-23





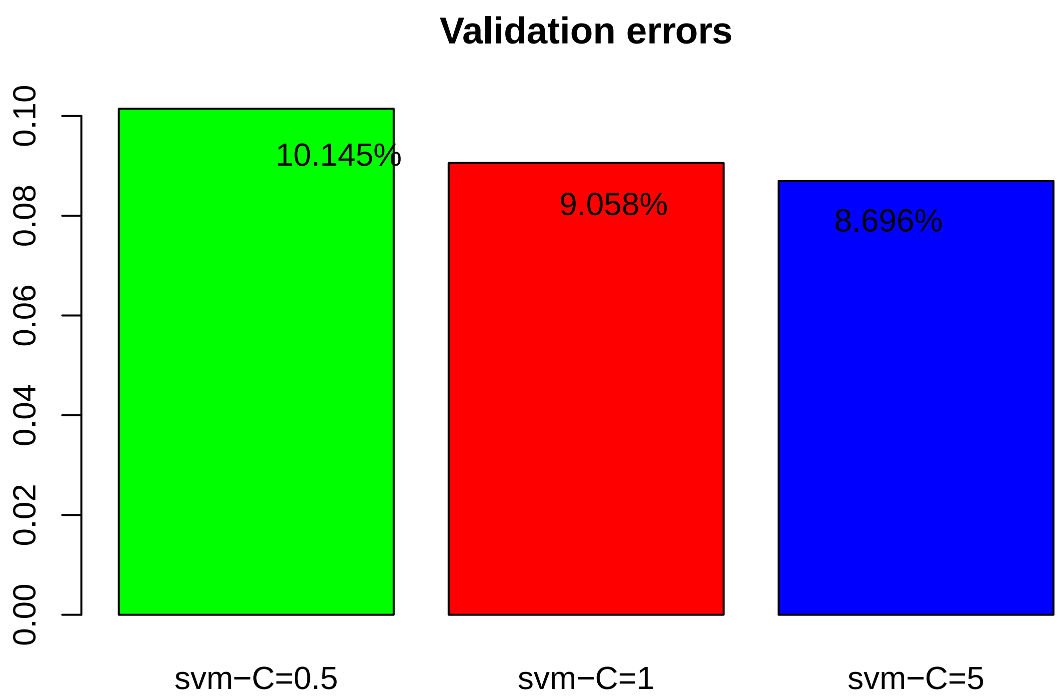
2018-12-23





When the distances d_i are smaller, the corresponding k_i would be larger depending on exponentiating the negative values. Therefore, a larger weight will be added into our prediction model. This is reasonable happened in "summation" model (5+0.1+0.1=5.2). However, these impacts might be smaller in "multiplication" model since multiplication can smaller the final kernel values if enough large data are considered (5*0.1*0.1=0.05). This might be the reason why "multiplication" model can always show us a wider range prediction than "summation" model.

Assignment 2



We choose the model with C=5 because it has the lowest misclassification error among the other values of C.Then we use the value of the parameter C in order to calculate the generalization error on a partion of the data we kept as test.

```
## ==============================================================
##  The generalization error for the best C value is:  7.600434 %
```

The model that will be returned to the user is:

```
## =========================================================

## 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 : 1547
##
## Objective Function Value : -2082.468
## Training error : 0.022169
```

Finally,the role of C is to control the regularization of the model.Once the C is large the model might have high variance. On the contrary a small C will lead to high bias for the model.

Contributions

Assignment 1 is from *Zijie Feng* and assignment 2 is from *Andreas Christopoulos Charitos*.

Appendix

```
knitr::opts_chunk$set(echo = TRUE, warning = FALSE)
#1.1##########################################################################
set.seed(1234567890)
library(geosphere)
Sys.setlocale(locale = "latin")
stations <- read.csv("stations.csv",stringsAsFactors = F,fileEncoding = "latin1")
temps <- read.csv("temps50k.csv",stringsAsFactors = F)
st <- merge(stations,temps,by="station_number")

times <- c("04:00:00", "06:00:00","08:00:00" ,"10:00:00","12:00:00","14:00:00",
           "16:00:00","18:00:00","20:00:00","22:00:00","24:00:00")

temp <- vector(length=length(times))
temp2 <- vector(length=length(times))

### Students’ code here #########################################################

h_distance <- 33000
h_date <-40
h_time <-4   # control the smoothness of temperatures in a day
a <- 58.4137
b <- 15.6235
date1 <- "2013-7-23"
# date1 <- "2013-12-23"

required<-st[st$date< date1,]   # only consider the dates before date1
times1 <- as.POSIXct(times,format="%H:%M:%S")

##distances by geography
d1<-distm(required[,c("latitude","longitude")], c(a,b))
k1<-exp(-(d1/h_distance)^2)

##distances by dates
d2<-as.numeric(
  difftime(as.POSIXct(date1,format="%Y-%m-%d"),
           as.POSIXct(required$date,format="%Y-%m-%d"),
           units="days")
)
k2<-exp(-(d2/h_date)^2)

for(i in 1:length(times1)){
  ##distances by hours
  dist <- as.numeric(
    difftime(as.POSIXct(required$time,format="%H:%M:%S"),
             times1[i],
             units="hours")
  )
  k3<-exp(-(dist/h_time)^2)
```

```r
  K<-as.vector(k1)+as.vector(k2)+as.vector(k3)
  temp[i]<-sum(K*required$air_temperature)/sum(K)
  K<-as.vector(k1)*as.vector(k2)*as.vector(k3)
  temp2[i]<-sum(K*required$air_temperature)/sum(K)

}

df <- data.frame(times=times,
                 temp=temp,
                 temp2=temp2)

library(ggplot2)
p1<-ggplot(df,aes(x=times, y=temp))+geom_point()+labs(title="summation")
p2<-ggplot(df,aes(x=times, y=temp2))+geom_point()+labs(title="multiplication")
plot(gridExtra::arrangeGrob(p1,p2))
knitr::include_graphics("7-23.png")
knitr::include_graphics("12-23.png")

#2#########################################################################
library(kernlab)

#load buid-in dataset from kernlab
data("spam")

#Split data to train,valid,test
n=dim(spam)[1]
set.seed(1234567890)
id=sample(1:n, floor(n*0.5))
train_spam=spam[id,]
id1=setdiff(1:n, id)
set.seed(1234567890)
id2=sample(id1, floor(n*0.3))
valid_spam=spam[id2,]
id3=setdiff(id1,id2)
test_spam=spam[id3,]

#fit svm models for every C value
svm1<-ksvm(type~.,data=train_spam,
           kernel = "rbfdot", kpar =list(sigma = 0.05),C = 0.5)

svm2<-ksvm(type~.,data=train_spam,
           kernel = "rbfdot", kpar =list(sigma = 0.05),C = 1)

svm3<-ksvm(type~.,data=train_spam,
           kernel = "rbfdot", kpar =list(sigma = 0.05),C = 5)


index_class_column<-which(names(spam)=="type") #index of the class column
#make predictions on valid data with the 3 models
svm1_preds_valid<-predict(svm1,valid_spam[,-index_class_column],
                          type="response")

svm2_preds_valid<-predict(svm2,valid_spam[,-which(names(spam)=="type")],
```

```
                           type="response")

svm3_preds_valid<-predict(svm3,valid_spam[,-index_class_column],
                           type="response")
#calculate misclassification errors on valid data
error1<-mean(valid_spam$type!=svm1_preds_valid)
error2<-mean(valid_spam$type!=svm2_preds_valid)
error3<-mean(valid_spam$type!=svm3_preds_valid)
#combine data
errors_valid<-c(error1,error2,error3)
#plot the misclassification errors
barplot(errors_valid,names.arg=c("svm-C=0.5", "svm-C=1", "svm-C=5"),col=c("green","red","blue"),
        main = "Validation errors")
text((errors_valid/1.1),labels=paste0(round(errors_valid*100,digits=3),"%"))

#combine train and valid data
dt<-rbind(train_spam,valid_spam)
#fit svm with train and valid data
svm_best<-ksvm(type~.,data=dt,
               kernel = "rbfdot", kpar =list(sigma = 0.05),C = 5)
#predict on test data
svm3_preds_test<-predict(svm_best,test_spam[,-58],type="response")
#calculate misclassification error
error6<-mean(test_spam$type!=svm3_preds_test)
cat("=============================================================\n",
  "The generalization error for the best C value is: ",error6*100,"%")

cat("======================================================\n")
svm_final<-ksvm(type~.,data=spam,
                kernel = "rbfdot", kpar =list(sigma = 0.05),C = 5)
svm_final
```