TDDE01. Lab3. Group B24 report.

# **Statement of contribution**

# **Assignment 2. Support Vector Machines**

We used Lab3Block1-2021-SVMs-St.R from lab material that performs SVM to classify the R spam data set by KSVM function. All SVM models use Gaussian kernel function (rbfdot) with the width of 0.05 and the C parameter is different in each model.

Four different filters were constructed with the C parameter which gives the lowest MCR.

**Task1:  Which filter do you return to the user? filter0, filter1, filter2 or filter3? Why?**

All filters are same except in data that they use in KSVM. You can see their differences in table below.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Filter 0** | **Filter 1** | **Filter 2** | **Filter 3** |
| **Data set in KSVM** | Data= Train | Data= Train | Data= Train+ Valid | Data= Spam |
| **Obs of 58 variables** | 3000 | 3000 | 3800 | 4601 |
| **Date set for Predict** | Data= valid | Data= Test | Data= Test | Data= Test |
| **Error** | 0.0675 | 0.08489388 | 0.082397 | 0.02122347 |
| **Accuracy** | 93.25 % | 91.51061 % | 91.7603 % | 97.87765 % |

Filter 0: Has smaller data sets than filter 2 and 3 and it does not use test data set for prediction.

Filter1: Such as filter 0, this filter also has smaller data sets than filter 2 and 3, but did not use the validation data set at all.

**Filter2: We prefer return this filter to user because it has the largest usage of available data while still keeping some separate for testing.**

Filter3: In this filter training dataset contained test dataset and it cannot be a good filter.

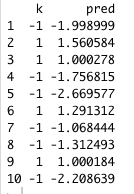
**Task2: What is the estimate of the generalization error of the filter returned to the user? err0, err1, err2 or err3? Why? err2**

To answer this question, we considered unseen data used, unseen data for C (reg term) and size of data set. At first step we reject err0, because both training and validation datasets were seen before. We reject err3 because it uses test data set in training.

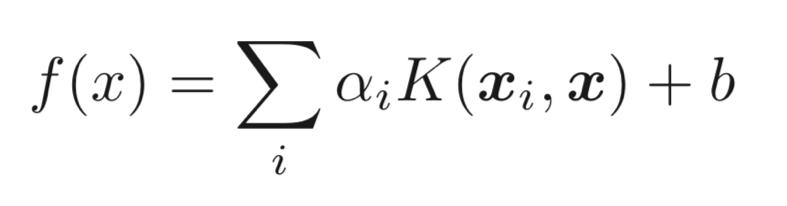
Between err1 or err 2, err1 is obtained on a smaller dataset (tr) and is a bit bigger than err2 while both of them use unseen data (which is test), so we prefer err2.

**Task3: Implementation of SVM predictions:**

In this task we are asked to implement the linear combination for filter 3. We used given code to get indexes of support vectors, linear coefficients for support vectors and the negative intercept. We produce prediction for the first 10 point of data set by a nested loop and compare it with the result of function predict where the type is "decision". As you can see the prediction results are identical.



We used those formulae to calculate our prediction while k is the RBF kernel:





Code:

# Lab 3 block 1 of 732A99/TDDE01/732A68 Machine Learning  
# Author: jose.m.pena@liu.se  
# Made for teaching purposes

library(kernlab)  
set.seed(1234567890)

data(spam)  
foo <- sample(nrow(spam))  
spam <- spam[foo,]  
spam[,-58]<-scale(spam[,-58])  
tr <- spam[1:3000, ]  
va <- spam[3001:3800, ]  
trva <- spam[1:3800, ]  
te <- spam[3801:4601, ]

by <- 0.3  
err\_va <- NULL  
for(i in seq(by,5,by)){  
  filter <- ksvm(type~.,data=tr,kernel="rbfdot",kpar=list(sigma=0.05),C=i,scaled=FALSE)  
  mailtype <- predict(filter,va[,-58])  
  t <- table(mailtype,va[,58])  
  err\_va <-c(err\_va,(t[1,2]+t[2,1])/sum(t))  
}

filter0 <- ksvm(type~.,data=tr,kernel="rbfdot",kpar=list(sigma=0.05),C=which.min(err\_va)\*by,scaled=FALSE)  
mailtype <- predict(filter0,va[,-58])  
t <- table(mailtype,va[,58])  
err0 <- (t[1,2]+t[2,1])/sum(t)  
err0

filter1 <- ksvm(type~.,data=tr,kernel="rbfdot",kpar=list(sigma=0.05),C=which.min(err\_va)\*by,scaled=FALSE)  
mailtype <- predict(filter1,te[,-58])  
t <- table(mailtype,te[,58])  
err1 <- (t[1,2]+t[2,1])/sum(t)  
err1

filter2 <- ksvm(type~.,data=trva,kernel="rbfdot",kpar=list(sigma=0.05),C=which.min(err\_va)\*by,scaled=FALSE)  
mailtype <- predict(filter2,te[,-58])  
t <- table(mailtype,te[,58])  
err2 <- (t[1,2]+t[2,1])/sum(t)  
err2

filter3 <- ksvm(type~.,data=spam,kernel="rbfdot",kpar=list(sigma=0.05),C=which.min(err\_va)\*by,scaled=FALSE)  
mailtype <- predict(filter3,te[,-58])  
t <- table(mailtype,te[,58])  
err3 <- (t[1,2]+t[2,1])/sum(t)  
err3

library(kernlab)

sv<-alphaindex(filter3)[[1]]

co<-coef(filter3)[[1]]

inte<- - b(filter3)

k<-0

rbfkernel <- rbfdot(sigma = 0.05)

kpar(rbfkernel)

for(i in 1:10){

k2<-0

for(j in 1:length(sv)){

x= as.vector(spam[sv[j],-58])

names(x)<-NULL

x<-as.numeric(x)

y= as.vector(spam[i,-58])

names(y)<-NULL

y<-as.numeric(y)

RBFkernel= rbfkernel(y,x)

k2 <- k2+co[j]\*RBFkernel

}

k2<-k2+inte

sig<-sign(k2)

k<-c(k,sig)

}

k=k[2:11]

pred=predict(filter3,spam[1:10,-58], type = "decision")

data.frame(k,pred)