# Untitled

*Prudhvi Peddmallu*

*17 December 2018*

The data file data.csv contains information about 64 e-mails which were manually collected from DBWorld mailing list. They were classified as: 'announces of conferences' (1) and 'everything else' (0) (variable Conference) 1. Divide data into training and test sets (70/30) without scaling. Perform nearest shrunken centroid classification of training data in which the threshold is chosen by cross-validation. Provide a centroid plot and interpret it. How many features were selected by the method? List the names of the 10 most contributing features and comment whether it is reasonable that they have strong effect on the discrimination between the conference mails and other mails? Report the test error. 2. Compute the test error and the number of the contributing features for the following methods fitted to the training data: a. Elastic net with the binomial response and ????????=0.5 in which penalty is selected by the cross-validation b. Support vector machine with "vanilladot" kernel. Compare the results of these models with the results of the nearest shrunken centroids (make a comparative table). Which model would you prefer and why? 3. Implement Benjamini-Hochberg method for the original data, and use t.test() for computing p-values. Which features correspond to the rejected hypotheses? Interpret the result.

```
library(pamr)
library(glmnet)
library(dplyr)
library(kernlab)
library(ggplot2)
library(akima)
library(mgcv)
library(readxl)
library(grid)
library(plotly)
library(dplyr)
```

## Question 2.1

## benjimani hosenberg and support vector machine.

```
# data_emails <- read.csv(file = "data.csv", sep = ";",header=T)
#
# data_emails$Conference=as.factor(data_emails$Conference)
# rownames(data_emails)=1:nrow(data_emails)
# #diveding the data
# n=dim(data_emails)[1]
# set.seed(12345)
# id=sample(1:n, floor(n*0.7))
# train=data_emails[id,]
# test=data_emails[-id,]
#
# x_train = t(train[,-4703])
# y_train = train[[4703]]
# x_test = t(test[,-4703])
```

```
# y_test = test[[4703]]
#
# mydata_e=list(x=x_train,y=as.factor(y_train),geneid=as.character(1:nrow(x_train)), genenames=rownames
# model = pamr.train(mydata_e,threshold=seq(0,4, 0.1))
# cvmodel=pamr.cv(model,mydata_e)
# pamr.plotcen(model, mydata_e, threshold=1.4)
# contri_g=pamr.listgenes(model,mydata_e,threshold=1.4)
#
# imp = as.numeric(contri_g[1:10,1])
# colna = colnames(data_emails)
# colna[imp]
#
#
# cat( paste( colnames(data_emails[,genesno_10$id]), collapse='\n' ) )
# #predection
# pre_test <- pamr.predict(model, newx = x_test, threshold=1.4)
# misclas_table <- table(y_test,pre_test)
# test_error <- 1 - sum(diag(misclas_table))/sum(misclas_table)
# test_error
```

## Question2.2

```
# x_train = train[,-4703]%>% as.matrix()
# y_train = train[[4703]]
# x_test = test[,-4703]%>% as.matrix()
# y_test = test[[4703]]
#
# cvfitt <- cv.glmnet(x_train, y_train, family = "binomial", alpha = 0.5)
# plot(cvfitt)
```

```
# prediction_elastic <- predict.cv.glmnet(cvfitt, newx = x_test, s="lambda.min",type = "class")
# coeffics <- coef(cvfitt,s="lambda.min")
# e_variables = as.data.frame(coeffics[which(coeffics[,1]!=0),])
# cmat_elastic <- table(y_test, prediction_elastic)
# teserror_elastic <- 1 -sum(diag(cmat_elastic))/sum(cmat_elastic)
# teserror_elastic
```

Test Error is- 0.1

```
# #SVM
# svm_modell <- ksvm(x_train, y_train, type = 'C-svc',  kernel = "vanilladot", scale = F)
# prediction_svm <- predict(svm_modell,newdata = x_test, type = "response")
# #table
# mtable_svm <- table(y_test,prediction_svm)
# teserror_svm <- 1 -sum(diag(mtable_svm))/sum(mtable_svm)
# teserror_svm
```

rejecting the null hypothesis(the features with p values less than 0.05).

```
# table_results <- data.frame("Model" = c("Nearest Shrunken Centroid Model", "ElasticNet Model", "SVM M
# "Error" = c(test_error, teserror_elastic,teserror_sum ))
# knitr::kable(table_results, caption = "Model Comparison")
```

## Question2.3-Implement Benjamini-Hochberg method

```
# p_values <- c()
# x <- data_emails[,-4703]
# #y <- as.vector(data[,4703])
# for(i in 1:(ncol(data_emails)-1)){
#
#   t_res <- t.test(x[,i]~Conference, data = data_emails)
#
#   p_values[i] <- t_res$p.value
#
# }
# p_adj <- p.adjust(p_values, method = "BH", n = length(p_values))
# p_adj_df <- data.frame("feature" = colnames(data_emails[,-4703]), "pvals" = p_adj)
# p_adj_df <- p_adj_df[which(p_adj_df[,2] <= 0.05), ]
# num_features <- nrow(p_adj_df)
```

```
# #features
# p_adj_df[c(1:39),1]
```

The above features are the significant features as selected by the model. From the above feature list, the words team, important, topics, presented, proceedings,salary ,candidate,held,degree,helg, org,international, call, papers, phd, published etc seems very relevant in classification of a conferance email.

# GAM AND GLM

## Question1

1. Use time series plots to visually inspect how the mortality and influenza number vary with time (use Time as X axis). By using this plot, comment how the amounts of influenza cases are related to mortality rates.
2. Use gam() function from mgcv package to fit a GAM model in which Mortality is normally distributed and modelled as a linear function of Year and spline function of Week, and make sure that the model parameters are selected by the generalized cross-validation. Report the underlying probabilistic model.
3. Plot predicted and observed mortality against time for the fitted model and comment on the quality of the fit. Investigate the output of the GAM model and report which terms appear to be significant in the model. Is there a trend in mortality change from one year to another? Plot the spline component and interpret the plot.
4. Examine how the penalty factor of the spline function in the GAM model from step 2 influences the estimated deviance of the model. Make plots of the predicted and observed mortality against time for cases of very high and very low penalty factors. What is the relation of the penalty factor to the degrees of freedom? Do your results confirm this relationship?
5. Use the model obtained in step 2 and plot the residuals and the influenza values against time (in one plot). Is the temporal pattern in the residuals correlated to the outbreaks of influenza? 6 Fit a GAM model in R in which mortality is be modelled as an additive function of the spline functions of year, week, and the number of confirmed cases of influenza. Use the output of this GAM function to conclude whether or not the mortality is influenced by the outbreaks of influenza. Provide the plot of the original and fitted Mortality against Time and comment whether the model seems to be better than the previous GAM models.

```
influenz<-read_xlsx("influenza.xlsx")
```

## Question1.1-Time series plots to visually inspect how the mortality and influenza

```
# #plots
# p<-ggplot(data = influenz, aes(x = Time, y = Influenza)) + geom_line(aes(color = "#00AFBB"))
# p
# q<-ggplot(data = influenz, aes(x = Time, y = Mortality)) + geom_line(aes(color = "#00AFBB"))
# q
```

There is increase in influenza cases there is increase in mortality till 2000. After that influenza seems to be having lesser impact on mortality.

## Question1.2-Fit a GAM model

```
# gam_model=gam(Mortality~Year+s(Week,k=length(unique(influenz$Week))),data=influenz,method="GCV.Cp")
# gam_model
```

## Question1.3

```
# fitt<-fitted(gam_model)
# influenz$predict_mortality<-fitt
# ggplot(influenz)+geom_line(aes(x=Time,y=Mortality,colour="red"))+geom_line(aes(x=Time,y=predict_morta
```

## Question1.4

```
# res_2 = gam(Mortality ~ Year +
#           s(Week, k=length(unique(influenz$Week)), sp=0),
#           method="GCV.Cp", data=influenz)
#
# predict_2 <- predict(res_2, newdata = influenz, type='response')
# p1 <- ggplot(data =influenz) +
#   geom_point(aes(Time,Mortality),colour = "Blue") +
#   geom_line(aes(Time,predict_2),colour = "Red")
#
# res_3 = gam(Mortality ~ Year +
#           s(Week, k=length(unique(influenz$Week)), sp=0.01),
#           method="GCV.Cp", data=influenz)
#
# predict_3 <- predict(res_3, newdata = influenz, type='response')
# p2 <- ggplot(data =influenz) +
#   geom_point(aes(Time,Mortality),colour = "Blue") +
#   geom_line(aes(Time,predict_3),colour = "Red")
#
# res_4 = gam(Mortality ~ Year +
#           s(Week, k=length(unique(influenz$Week)), sp=1),
#           method="GCV.Cp", data=influenz)
```

```
# predict_4 <- predict(res_4, newdata = influenz, type='response')
# p3 <- ggplot(data =influenz) +
#   geom_point(aes(Time,Mortality),colour = "Blue") +
#   geom_line(aes(Time,predict_4),colour = "Red")
#
# grid.newpage()
# grid.draw(cbind(ggplotGrob(p1), ggplotGrob(p2), ggplotGrob(p3), size = "last"))
#
# anova(res_2)
# anova(res_3)
# anova(res_4)
#
# #some more models with different sp values
# res_5 = gam(Mortality ~ Year +
#           s(Week, k=length(unique(influenz$Week)), sp=1.5),
#           method="GCV.Cp", data=influenz)
#
# res_6 = gam(Mortality ~ Year +
#           s(Week, k=length(unique(influenz$Week)), sp=2),
#           method="GCV.Cp", data=influenz)
#
# res_7 = gam(Mortality ~ Year +
#           s(Week, k=length(unique(influenz$Week)), sp=3),
#           method="GCV.Cp", data=influenz)
#
# res_8 = gam(Mortality ~ Year +
#           s(Week, k=length(unique(influenz$Week)), sp=100),
#           method="GCV.Cp", data=influenz)
#
# #edf vs sp
# x <- c(0, 0.01, 1, 1.5, 2, 3, 100)
# y <- c(sum(res_2$edf), sum(res_3$edf), sum(res_4$edf), sum(res_5$edf), sum(res_6$edf), sum(res_7$edf)
# data.frame(cbind(x,y))
```

## Question1.5

```
# residuals<-gam_model$residuals
# influenz$Residuals<-residuals
# ggplot(influenz)+geom_line(aes(x=Time,y=Residuals,color="red"))+geom_line(aes(x=Time,y=Influenza,colo
```

## Question1.6

```
# gam_model1=gam(Mortality~s(Year,k=length(unique(influenz$Year)))+s(Week,k=length(unique(influenz$Week
# gam_model1
```

```
# fitt1<-fitted(gam_model1)
# influenz$predict_mortality1<-fitt1
# ggplot(influenz)+geom_line(aes(x=Time,y=Mortality,colour="red"))+geom_line(aes(x=Time,y=predict_morta
```