Lab2Block2

Prudhvi Peddmallu

22 April 2019

Assignment 1. Using GAM and GLM to examine the mortality rates

The Excel document influenza.xlsx contains weekly data on the mortality and the number of laboratory-confirmed cases of influenza in Sweden. In addition, there is information about population-weighted temperature anomalies (temperature deficits).

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.

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

```r
#data-file
library(xlsx)
flu_data = read.xlsx("influenza.xlsx", sheetName = "Raw data")
library(readxl)
influenza<-read_xlsx("influenza.xlsx")
```

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

```
library(ggplot2)
#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$Week))),data=influenz,method="GCV.Cp")
gam_model
(or)
res = gam(Mortality ~ Year +s(Week, k=length(unique(influenza$Week))),method="GCV.Cp", data=influenza)
gam.check(res)
(or)
gam_model <- mgcv::gam(data = flu_data, Mortality~Year+s(Week), method = "GCV.Cp")
summary(gam_model)
```

Analysis: Using the default parameter settings within the *gam*-function implies that *Mortality* is normally distributed (*family=gaussian()*). Also, since *method = "GCV.Cp"*, this leads to the usage of GCV (*Generalized Cross Validation score*) related to the smoothing parameter estimation. The underlying probabilistic model can be written as:

$$Mortality = N(\mu, \sigma^2)$$

$$\hat{Mortality} = Intercept + \beta_1 Year + s(Week) + \epsilon$$

where

$$\epsilon = N(0, \sigma^2).$$

Question1.3:-Plot predicted and observed mortality against time for the fitted model and comment on the quality of the fit.

```
predicted <- predict(res, newdata = influenza, type='response')
ggplot(data =influenza) +
geom_point(aes(Time,Mortality), colour = "Blue") +
geom_line(aes(Time,predicted), colour = "Red")
summary(res)
plot(res)#Plot the spline component
```

Analysis:Significant terms in model - According to the p values in the summary, Spline function of weekis the most sognificant term.Trend in mortality change from one year to another - It seems to be following a cyclic trendand seems to have decreased with the years.

Question1.4:-the penalty factor of the spline function in the GAM model

```r
res2 = gam(Mortality ~ Year +
s(Week, k=length(unique(influenza$Week)), sp=0),
method="GCV.Cp", data=influenza)
predicted2 <- predict(res2, newdata = influenza, type='response')
p1 <- ggplot(data =influenza) +
geom_point(aes(Time,Mortality),colour = "Blue") +
geom_line(aes(Time,predicted2),colour = "Red") +
ggtitle("sp = 0")
res3 = gam(Mortality ~ Year +
s(Week, k=length(unique(influenza$Week)), sp=0.01),
method="GCV.Cp", data=influenza)
predicted3 <- predict(res3, newdata = influenza, type='response')
p2 <- ggplot(data =influenza) +
geom_point(aes(Time,Mortality),colour = "Blue") +
geom_line(aes(Time,predicted3),colour = "Red") +
ggtitle("sp = 0.01")
res4 = gam(Mortality ~ Year +
s(Week, k=length(unique(influenza$Week)), sp=1),
method="GCV.Cp", data=influenza)
predicted4 <- predict(res4, newdata = influenza, type='response')
p3 <- ggplot(data =influenza) +
geom_point(aes(Time,Mortality),colour = "Blue") +
geom_line(aes(Time,predicted4),colour = "Red") +
ggtitle("sp = 1")
grid.newpage()
grid.draw(cbind(ggplotGrob(p1), ggplotGrob(p2), ggplotGrob(p3), size = "last"))
#anova(res2)
#some more models with different sp values
res5 = gam(Mortality ~ Year +
s(Week, k=length(unique(influenza$Week)), sp=1.5),
method="GCV.Cp", data=influenza)
res6 = gam(Mortality ~ Year +
s(Week, k=length(unique(influenza$Week)), sp=2),method="GCV.Cp", data=influenza)
res7 = gam(Mortality ~ Year +
s(Week, k=length(unique(influenza$Week)), sp=3),
method="GCV.Cp", data=influenza)
res8 = gam(Mortality ~ Year +
s(Week, k=length(unique(influenza$Week)), sp=6),
method="GCV.Cp", data=influenza)
res9 = gam(Mortality ~ Year +
s(Week, k=length(unique(influenza$Week)), sp=9),
method="GCV.Cp", data=influenza)
res10 = gam(Mortality ~ Year +
s(Week, k=length(unique(influenza$Week)), sp=100),
method="GCV.Cp", data=influenza)
2-method
model_deviance <- NULL
for(sp in c(0.001, 0.01, 0.1, 1, 10))
{
k=length(unique(flu_data$Week))

gam_model <- mgcv::gam(data = flu_data, Mortality~Year+s(Week, k=k, sp=sp), method = "GCV.Cp")
temp <- cbind(gam_model$deviance, gam_model$fitted.values, gam_model$y, flu_data$Time_fixed,
              sp, sum(influence(gam_model)))
model_deviance <- rbind(temp, model_deviance)
}
model_deviance <- as.data.frame(model_deviance)
colnames(model_deviance) <- c("Deviance", "Predicted_Mortality", "Mortality", "Time",
                              "penalty_factor", "degree_of_freedom")
model_deviance$Time <- as.Date(model_deviance$Time, origin = '1970-01-01')
# plot of deviance
p6 <- ggplot(data=model_deviance, aes(x = penalty_factor, y = Deviance)) +
geom_point() +
  geom_line() +
      theme_light() +
ggtitle("Plot of Deviance of Model vs. Penalty Factor")
p6
# plot of degree of freedom
p7 <- ggplot(data=model_deviance, aes(x = penalty_factor, y = degree_of_freedom)) +
geom_point() +
  geom_line() +
      theme_light() +
ggtitle("Plot of degree_of_freedom of Model vs. Penalty Factor")
p7
model_deviance_wide <- melt(model_deviance[,c("Time", "penalty_factor",
                                              "Mortality", "Predicted_Mortality")],
                            id.vars = c("Time", "penalty_factor"))
# plot of predicted vs. observed mortality
p8 <- ggplot(data=model_deviance_wide[model_deviance_wide$penalty_factor == 0.001,],
             aes(x= Time, y = value)) +
  geom_point(aes(color = variable), size=0.7) +
  geom_line(aes(color = variable), size=0.7) +
  scale_color_manual(values=c("#E69F00", "#009E73")) +
  theme_light() +
  ggtitle("Plot of Mortality vs. Time(Penalty 0.001)")
p9 <- ggplot(data=model_deviance_wide[model_deviance_wide$penalty_factor == 10,],
             aes(x= Time, y = value)) +
  geom_point(aes(color = variable), size=0.7) +
    geom_line(aes(color = variable), size=0.7) +
  scale_color_manual(values=c("#E69F00", "#009E73")) +
    theme_light() +
  ggtitle("Plot of Mortality vs. Time(Penalty 10)")
p8
p9
```

Question1.5:-plot the residuals and the influenza values against time (in one plot).

```
edf is estimated degree of freedom if you see res2 u can get edf
edf vs sp
x <- c(0, 0.01, 1, 1.5, 2, 3, 6, 9, 100)
y <- c(sum(res2$edf), sum(res3$edf), sum(res4$edf), sum(res5$edf), sum(res6$edf),
sum(res7$edf), sum(res8$edf), sum(res9$edf), sum(res10$edf))
data.frame(cbind(sp=x,edf=y) )
ggplot(data = influenza, aes(x = Time)) +
geom_line(aes(y = Influenza,colour = "Influenza")) +
geom_line(aes(y = res$residuals,colour = "Residuals")) +
scale_colour_manual("", breaks = c("Influenza", "Residuals"),
values = c("Red", "Blue"))
result
# sp edf
# 1 0.00 29.000000
# 2 0.01 6.689771
# 3 1.00 3.491752
# 4 1.50 3.365887
# 5 2.00 3.291922
# 6 3.00 3.208210
# 7 6.00 3.112130
# 8 9.00 3.076759
# 9 100.00 3.007268
#1.5
k=length(unique(flu_data$Week))
gam_model <- mgcv::gam(data = flu_data, Mortality~Year+s(Week, k=k), method = "GCV.Cp")
temp <- flu_data
temp <- cbind(temp, residuals = gam_model$residuals)
p10 <- ggplot(data = temp, aes(x = Time_fixed)) +
  geom_line(aes( y = Influenza, color = "Influenza")) +
  geom_line(aes(y = residuals, color = "residuals")) +
      theme_light() +
  scale_color_manual(values=c(Influenza = "#009E73", residuals = "#E69F00")) +
  labs(y = "Influenza / Residual") +
  ggtitle("Plot of Influenza Residual vs. Time")
p10
```

Analysis:With increase in Penalty factor the degrees of freedom should decrease. Our results also confirm this relationship. Initially the decline is steep. Some of the peaks in Influenza outbreaks correspond to peaks in the residuals of the fitted model. Still,however, a lot of variance in the residuals is not correlated to Influenza outbreaks. Therefore, I would say that the Influenza outbreaks are not correlated to the residuals.

Question1.6:-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.

```
new_res=gam(Mortality~s(Year,k=length(unique(influenza$Year)))
+s(Week, k=length(unique(influenza$Year))))this is continued up
summary(new_res)
plot(new_res, residuals=TRUE, page=1)
new_predicted <- predict(new_res, newdata = influenza, type='response')
ggplot(data = influenza, aes(x = Time)) +
geom_point(aes(y = Mortality, colour = "Mortality")) +
geom_line(aes(y = new_predicted,colour = "New Model")) +
geom_line(aes(y = predicted,colour = "Previous Model")) +
scale_colour_manual("", breaks = c("Mortality","New Model", "Previous Model"),
values = c("Blue", "Red", "Green"))
#(or)
gam_model_additive <- mgcv::gam(data = flu_data, Mortality~s(Year)+s(Week), method = "GCV.Cp")
k1 = length(unique(flu_data$Year))
```

```
k2 = length(unique(flu_data$Week))
k3 = length(unique(flu_data$Influenza))
gam_model_additive <- gam(Mortality ~ s(Year, k=k1) +
                                     s(Week, k=k2) +
                                    s(Influenza, k=k3),
                          data = flu_data)
summary(gam_model_additive)
flu_data$fitted.values = gam_model_additive$fitted.values

p11 <- ggplot(data = flu_data, aes(x = Time_fixed)) +
  geom_line(aes( y = Mortality, color = "Mortality")) +
  geom_line(aes(y = fitted.values, color = "fitted.values")) +
      theme_light() +
  scale_color_manual(values=c(Mortality = "#009E73", fitted.values = "#E69F00")) +
  labs(y = "Mortality / fitted.values") +
  ggtitle("Plot of Mortality and Fitted vs. Time")
p11
```

Analysis:-From the p values, we can conclude that Influenza cases have high impact on mortality The additive GAM model clearly has the best fit. Much of the variance of the data (81.9% is the adjusted R^2) is captured by the model. Given that the GAM models in step 2 and step 4 do not include the influenza variable from the dataset, and the the model above does, one can say that most likely mortality is influenced by the outbreaks of influenza.

Assignment 2. High-dimensional methods

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 α ????????=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

Question2.1:- Perform nearest shrunken centroid classification

```
library(pamr)
library(glmnet)
library(dplyr)
library(kernlab)
# dividing data into train and test set
data <- read.csv(file = "data.csv", sep = ";", header = TRUE, fileEncoding = "Latin1")
data$Conference=as.factor(data$Conference)
rownames(data)=1:nrow(data)
n=dim(data)[1]
```

```r
set.seed(12345)
id=sample(1:n, floor(n*0.7))#0.7 is 70%
train=data[id,]
test=data[-id,]
#Perform of training data in which the threshold is chosen
# by cross-validation
x = t(train[,-4703])#4703 are the variables in the data
y = train[[4703]]
x_test = t(test[,-4703])
y_test = test[[4703]]
mydata=list(x=x,y=as.factor(y),geneid=as.character(1:nrow(x)), genenames=rownames(x))
model = pamr.train(mydata,threshold=seq(0,4, 0.1))
cvmodel=pamr.cv(model,mydata)
pamr.plotcv(cvmodel)
# from cv model,when the threshold is 1.3 and 1.4, the error is least. Hence selecting the threshold as
pamr.plotcen(model, mydata, threshold=1.4)
# The method selected 231 features
contri_genes=pamr.listgenes(model,mydata,threshold=1.4)
# 10 most contributing features
imp = as.numeric(contri_genes[1:10,1])
cat( paste( colnames(data[imp]), collapse='\n' ) )
#test error
pred_test <- pamr.predict(model, newx = x_test, threshold=1.4)
misclass_table <- table(y_test,pred_test)
test_error <- 1 - sum(diag(misclass_table))/sum(misclass_table)
#Elastic net
x = train[,-4703]%>% as.matrix()
y = train[[4703]]
x_test = test[,-4703]%>% as.matrix()
y_test = test[[4703]]
cvfit <- cv.glmnet(x, y, family = "binomial", alpha = 0.5)
plot(cvfit)
predict_elastic <- predict.cv.glmnet(cvfit, newx = x_test, s="lambda.min",type = "class")
coeffs <- coef(cvfit,s="lambda.min")
e_variables = as.data.frame(coeffs[which(coeffs[,1]!=0),])
cmatrix_elastic <- table(y_test, predict_elastic)
testerror_elastic <- 1 -sum(diag(cmatrix_elastic))/sum(cmatrix_elastic)
```

From plot of cross validation model, which shows the depandance of misclassification error vs the threshold value, we can see that the misclassification error is least when the threshold is 1.3 and 1.4. So for this analysis, threshold value of 1.4 is chosen. Nearest centroid method computes a standardized centroid for each class. The nearest shrunken centroid classification "shrinks" the class centroids towards zero by threshold, setting it equal to zero if it hits zero. After the shrinkage, the new sample is classifies using the centroid rule. So this will make the classifier more accurate by reducing the effecte of unimportant features so feature selection happens here. Here the centroid plot shows the shrunken centroids for each of the variables which has non zero differance to the shrunken centroids of two classes. The method selected 231 features. The 10 most contributing features are: paper, important, submissions, due, published, position, call, conferance, dates, candidates All the names and features mentioned above are significant in classification of emails as conferance emails or not. Test Error : 0.1 - 10% test error and 90% accuracy.

Question2.2:-SVM

```
#SVM
svm_model <- ksvm(x, y, type = 'C-svc', kernel = "vanilladot", scale = F)
predict_svm <- predict(svm_model,newdata = x_test, type = "response")
mtable_svm <- table(y_test,predict_svm)
testerror_svm <- 1 -sum(diag(mtable_svm))/sum(mtable_svm)
# table
table_result <- data.frame("Model" = c("Nearest Shrunken Centroid Model",
"ElasticNet Model", "SVM Model"), "Features" = c(231,40,41),
"Error" = c(test_error, testerror_elastic,testerror_svm ))
knitr::kable(table_result, caption = "Model Comparison")
```

On comparing all three models, Nearest Shrunken Centeroid model and Elastic model has same rate of test error which is 0.1 (10%) where centeroid model takes 231 parameters as contributing parameters whereas Elastic net model makes use of less number of features to do the prediction as the contributing features. In case of SVM model, it uses 41 features as contributing features and the error rate is 0.05 or 5% which is less than the above two models and it uses 41 features as contributing features which is not significantly differant compared to the elastic model. So based on the above two factors, SVM model is better for classification of the data because of the accuracy that the model provides with less number of features

Question:-2.3-Benjamini-Hochberg method

```
pvalues <- c()
x <- data[,-4703]
#y <- as.vector(data[,4703])
for(i in 1:(ncol(data)-1)){
t_res <- t.test(x[,i]~Conference, data = data)
14
pvalues[i] <- t_res$p.value
}
p_adj <- p.adjust(pvalues, method = "BH", n = length(pvalues))
p_adj_df <- data.frame("feature" = colnames(data[,-4703]), "pvals" = p_adj)
p_adj_df <- p_adj_df[which(p_adj_df[,2] <= 0.05), ]
num_features <- nrow(p_adj_df)
p_adj_df[c(1:39),1]
```

We consideres null hypothesis as the feature is not significant for analysing if the email is a conferance email and alternate hopothesis: The feature is significant in classifying coferance email. From the above analysis, we can see that the model selected 39 features as significant features by rejecting the null hypothesis(the features with p values less than 0.05). The features corresponding to rejected hypothesis are: [1] apply authors call camera candidate candidates chairs ## [36] submission team topics workshop 4702 Levels: a4soa aaai aachen aalborg aamas aarhus aaron aau abbadi abdalla abdallah abductive abilities The above features are the significant features as selected by the model. From the above feature list, the words team, important, topics, presented, proceedings, helg, org, international, call, papers, phd, published etc seems very relevant in classification of a conferance email

Assignment 2-2method

2.1.

```
rm(list=ls())
gc()
data <- read.csv(file = "data.csv", sep = ";", header = TRUE)
```

```
n=NROW(data)
data$Conference <- as.factor(data$Conference)
set.seed(12345)
id=sample(1:n, floor(n*0.7))
train=data[id,]
test = data[-id,]
rownames(train)=1:nrow(train)
x=t(train[,-4703])
y=train[[4703]]
rownames(test)=1:nrow(test)
x_test=t(test[,-4703])
y_test=test[[4703]]
mydata = list(x=x,y=as.factor(y),geneid=as.character(1:nrow(x)), genenames=rownames(x))
mydata_test = list(x=x_test,y=as.factor(y_test),geneid=as.character(1:nrow(x)), genenames=rownames(x))
model=pamr.train(mydata,threshold=seq(0, 4, 0.1))
cvmodel=pamr.cv(model, mydata)
important_gen <- as.data.frame(pamr.listgenes(model, mydata, threshold = 1.3))
predicted_scc_test <- pamr.predict(model, newx = x_test, threshold = 1.3)
```

```
### plots
pamr.plotcv(cvmodel)
pamr.plotcen(model, mydata, threshold = 1.3)
```

```
### important features
## List the significant genes
NROW(important_gen)
temp <- colnames(data) %>% as.data.frame()
colnames(temp) <- "col_name"
temp$index <- row.names(temp)
df <- merge(x = important_gen, y = temp, by.x = "id", by.y = "index", all.x = TRUE)
df <- df[order(df[,3], decreasing = TRUE ),]
#knitr::kable(head(df[,4],10), caption = "Important feaures selected by Nearest Shrunken Centroids ")
```

```
### confusion table
library(caret)
conf_scc <- table(y_test, predicted_scc_test)
names(dimnames(conf_scc)) <- c("Actual Test", "Predicted Srunken Centroid Test")
result_scc <- caret::confusionMatrix(conf_scc)
#caret::confusionMatrix(conf_scc)
```

2.2

```
x = train[,-4703] %>% as.matrix()
y = train[,4703]
```

```
x_test = test[,-4703] %>% as.matrix()
y_test = test[,4703]
cvfit = cv.glmnet(x=x, y=y, alpha = 0.5, family =   "binomial")
predicted_elastic_test <- predict.cv.glmnet(cvfit, newx = x_test, s = "lambda.min", type = "class")
tmp_coeffs <- coef(cvfit, s = "lambda.min")
elastic_variable <- data.frame(name = tmp_coeffs@Dimnames[[1]][tmp_coeffs@i + 1], coefficient = tmp_coe
#knitr::kable(elastic_variable, caption = "Contributing features in the elastic model")
conf_elastic_net <- table(y_test, predicted_elastic_test)
names(dimnames(conf_elastic_net)) <- c("Actual Test", "Predicted ElasticNet Test")
result_elastic_net <- caret::confusionMatrix(conf_elastic_net)
#caret::confusionMatrix(conf_elastic_net)
# svm
svm_fit <- kernlab::ksvm(x, y, kernel="vanilladot", scale = FALSE, type = "C-svc")
predicted_svm_test <- predict(svm_fit, x_test, type="response")
conf_svm_tree <- table(y_test, predicted_svm_test)
names(dimnames(conf_svm_tree)) <- c("Actual Test", "Predicted SVM Test")
result_svm <- caret::confusionMatrix(conf_svm_tree)
#caret::confusionMatrix(conf_svm_tree)
# creating table
final_result <- cbind(result_scc$overall[[1]]*100,
                      result_elastic_net$overall[[1]]*100,
                      result_svm$overall[[1]] *100) %>% as.data.frame()
features_count <- cbind(NROW(important_gen), NROW(elastic_variable), NCOL(data))
final_result <- rbind(final_result, features_count)
colnames(final_result) <- c("Nearest Shrunken Centroid Model",
                            "ElasticNet Model", "SVM Model")
rownames(final_result) <- c("Accuracy", "Number of Features")
#knitr::kable(final_result, caption = "Comparsion of Models on Test dataset")
```

2.3. 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.

```
y <- as.factor(data[,4703])
x <- as.matrix(data[,-4703])
p_values <- data.frame(feature = '',P_value = 0,stringsAsFactors = FALSE)
for(i in 1:ncol(x)){
res = t.test(x[,i]~y, data = data,
alternative="two.sided"
,conf.level = 0.95)
p_values[i,] <- c(colnames(x)[i],res$p.value)
}
p_values$P_value <- as.numeric(p_values$P_value)
p <- p.adjust(p_values$P_value, method = 'BH')
length(p[which(p > 0.05)])
out <- p_values[which(p <= 0.05),]
out <- out[order(out$P_value),]
rownames(out) <- NULL
#out
```