Homework 3

Preston Robertson

12/7/2021

library(pacman)

## Warning: package 'pacman' was built under R version 4.1.2

require(caTools)

## Loading required package: caTools

## Warning: package 'caTools' was built under R version 4.1.2

library(glmnet)

## Warning: package 'glmnet' was built under R version 4.1.2

## Loading required package: Matrix

## Loaded glmnet 4.1-3

library(dplyr)

##   
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':  
##   
## filter, lag

## The following objects are masked from 'package:base':  
##   
## intersect, setdiff, setequal, union

library(psych)

## Warning: package 'psych' was built under R version 4.1.2

library(tidyverse)

## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --

## v ggplot2 3.3.5 v purrr 0.3.4  
## v tibble 3.1.5 v stringr 1.4.0  
## v tidyr 1.1.4 v forcats 0.5.1  
## v readr 2.1.0

## Warning: package 'readr' was built under R version 4.1.2

## -- Conflicts ------------------------------------------ tidyverse\_conflicts() --  
## x ggplot2::%+%() masks psych::%+%()  
## x ggplot2::alpha() masks psych::alpha()  
## x tidyr::expand() masks Matrix::expand()  
## x dplyr::filter() masks stats::filter()  
## x dplyr::lag() masks stats::lag()  
## x tidyr::pack() masks Matrix::pack()  
## x tidyr::unpack() masks Matrix::unpack()

library(caret)

## Loading required package: lattice

##   
## Attaching package: 'caret'

## The following object is masked from 'package:purrr':  
##   
## lift

library(pls)

##   
## Attaching package: 'pls'

## The following object is masked from 'package:caret':  
##   
## R2

## The following object is masked from 'package:stats':  
##   
## loadings

library(e1071)  
library(MASS)

##   
## Attaching package: 'MASS'

## The following object is masked from 'package:dplyr':  
##   
## select

library(klaR)  
library(nnet)  
library(sigmoid)

##   
## Attaching package: 'sigmoid'

## The following object is masked from 'package:e1071':  
##   
## sigmoid

## The following objects are masked from 'package:psych':  
##   
## logistic, logit

library(nnet)  
library(fastDummies)  
library(gam); data("kyphosis")

## Loading required package: splines

## Loading required package: foreach

##   
## Attaching package: 'foreach'

## The following objects are masked from 'package:purrr':  
##   
## accumulate, when

## Loaded gam 1.20

head(kyphosis)

## Kyphosis Age Number Start  
## 1 absent 71 3 5  
## 2 absent 158 3 14  
## 3 present 128 4 5  
## 4 absent 2 5 1  
## 5 absent 1 4 15  
## 6 absent 1 2 16

#### Question 1:

#### Use data set kyphosis from R package “gam”, randomly divide the data into training (75%) and testing (25%) set.

#Splitting Data into Training and Testing Data  
set.seed(11)  
  
training.samples <- kyphosis$Kyphosis %>%  
 createDataPartition(p = 0.75, list = FALSE)  
train.data <- kyphosis[training.samples, ]  
train.data = data.frame(train.data)  
test.data <- kyphosis[-training.samples, ]  
test.data = data.frame(test.data)  
  
  
#Proving the proper split  
SizeOriginal <- nrow(kyphosis)  
SizeTraining <- nrow(train.data)  
SizeTest <- nrow(test.data)  
PercentageTraining <- (SizeTraining/SizeOriginal)\*100  
PercentageTest <- (SizeTest/SizeOriginal)\*100  
  
sprintf('Percent of Testing Data = %f', PercentageTest)

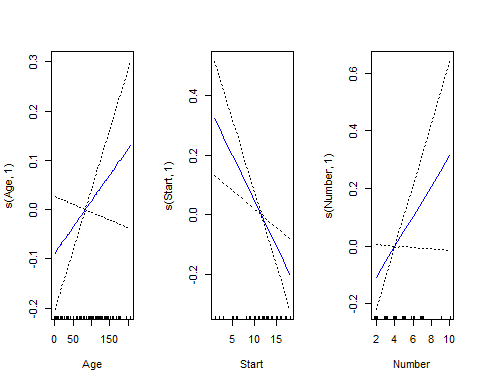
## [1] "Percent of Testing Data = 24.691358"

sprintf('Percent of Training Data = %f', PercentageTraining)

## [1] "Percent of Training Data = 75.308642"

#### Question 1.1: Use function “gam” to fit a logistic regression model. Add nonlinear terms as you see fit.

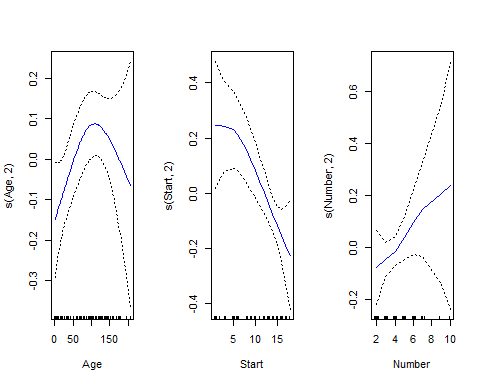
gam2 = gam(I(Kyphosis == "present") ~ s(Age, 1) + s(Start, 1) + s(Number, 1), data = kyphosis)  
par(mfrow = c(1,3))  
plot(gam2, se = TRUE, col = "blue")



summary(gam2)

##   
## Call: gam(formula = I(Kyphosis == "present") ~ s(Age, 1) + s(Start,   
## 1) + s(Number, 1), data = kyphosis)  
## Deviance Residuals:  
## Min 1Q Median 3Q Max   
## -0.79440 -0.22356 -0.08478 0.10205 0.84767   
##   
## (Dispersion Parameter for gaussian family taken to be 0.1295)  
##   
## Null Deviance: 13.4321 on 80 degrees of freedom  
## Residual Deviance: 9.9738 on 77 degrees of freedom  
## AIC: 70.2143   
##   
## Number of Local Scoring Iterations: NA   
##   
## Anova for Parametric Effects  
## Df Sum Sq Mean Sq F value Pr(>F)   
## s(Age, 1) 1 0.2148 0.21482 1.6585 0.20166   
## s(Start, 1) 1 2.7688 2.76875 21.3755 1.495e-05 \*\*\*  
## s(Number, 1) 1 0.4747 0.47473 3.6651 0.05928 .   
## Residuals 77 9.9738 0.12953   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Anova for Nonparametric Effects  
## Npar Df Npar F Pr(F)   
## (Intercept)   
## s(Age, 1) 0 5.3410 0.0001229 \*\*\*  
## s(Start, 1) 0 2.0050 1.855e-05 \*\*\*  
## s(Number, 1) 0 1.4719 6.649e-07 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

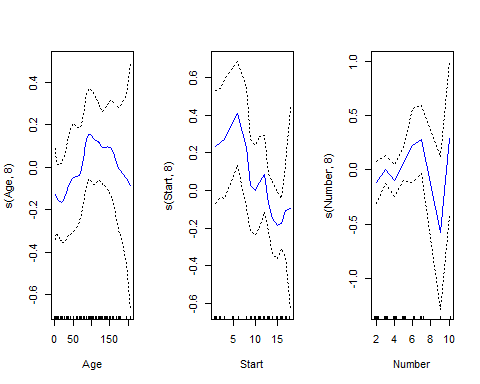
gam3 = gam(I(Kyphosis == "present") ~ s(Age, 2) + s(Start, 2) + s(Number, 2), data = kyphosis)  
par(mfrow = c(1,3))  
plot(gam3, se = TRUE, col = "blue")



summary(gam3)

##   
## Call: gam(formula = I(Kyphosis == "present") ~ s(Age, 2) + s(Start,   
## 2) + s(Number, 2), data = kyphosis)  
## Deviance Residuals:  
## Min 1Q Median 3Q Max   
## -0.71851 -0.21284 -0.08937 0.13918 0.83742   
##   
## (Dispersion Parameter for gaussian family taken to be 0.1232)  
##   
## Null Deviance: 13.4321 on 80 degrees of freedom  
## Residual Deviance: 9.1148 on 74 degrees of freedom  
## AIC: 68.9197   
##   
## Number of Local Scoring Iterations: NA   
##   
## Anova for Parametric Effects  
## Df Sum Sq Mean Sq F value Pr(>F)   
## s(Age, 2) 1 0.2180 0.21797 1.7697 0.1875   
## s(Start, 2) 1 2.5198 2.51980 20.4573 2.276e-05 \*\*\*  
## s(Number, 2) 1 0.3167 0.31668 2.5710 0.1131   
## Residuals 74 9.1148 0.12317   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Anova for Nonparametric Effects  
## Npar Df Npar F Pr(F)   
## (Intercept)   
## s(Age, 2) 1 4.3134 0.04128 \*  
## s(Start, 2) 1 1.9533 0.16641   
## s(Number, 2) 1 0.7029 0.40451   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

gam4 = gam(I(Kyphosis == "present") ~ s(Age, 8) + s(Start, 8) + s(Number, 8), data = kyphosis)  
par(mfrow = c(1,3))  
plot(gam4, se = TRUE, col = "blue")



summary(gam4)

##   
## Call: gam(formula = I(Kyphosis == "present") ~ s(Age, 8) + s(Start,   
## 8) + s(Number, 8), data = kyphosis)  
## Deviance Residuals:  
## Min 1Q Median 3Q Max   
## -0.56001 -0.19540 -0.02147 0.12652 0.79512   
##   
## (Dispersion Parameter for gaussian family taken to be 0.1223)  
##   
## Null Deviance: 13.4321 on 80 degrees of freedom  
## Residual Deviance: 6.9723 on 57.0001 degrees of freedom  
## AIC: 81.2155   
##   
## Number of Local Scoring Iterations: NA   
##   
## Anova for Parametric Effects  
## Df Sum Sq Mean Sq F value Pr(>F)   
## s(Age, 8) 1 0.4231 0.42311 3.459 0.06807 .   
## s(Start, 8) 1 3.0664 3.06641 25.068 5.658e-06 \*\*\*  
## s(Number, 8) 1 0.2764 0.27644 2.260 0.13828   
## Residuals 57 6.9723 0.12232   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Anova for Nonparametric Effects  
## Npar Df Npar F Pr(F)  
## (Intercept)   
## s(Age, 8) 7 0.69659 0.6746  
## s(Start, 8) 7 1.14700 0.3475  
## s(Number, 8) 6 1.36658 0.2438

#### Question 1.2: Compare your testing results of the model built in 1.1 with the results you obtained in HW2 Q2 (2.2). Do you observe any improvements? Why or why not?

# Rerun of Homework 2 Question 2  
  
# Fit the model  
model.lReg <- nnet::multinom(Kyphosis ~., data = train.data)

## # weights: 5 (4 variable)  
## initial value 42.281978   
## iter 10 value 25.720280  
## final value 25.720273   
## converged

predicted.lReg <- model.lReg %>% predict(test.data)  
accuracy.lReg <- mean(predicted.lReg == test.data$Kyphosis)  
sprintf('Model Accuracy = %f', accuracy.lReg)

## [1] "Model Accuracy = 0.850000"

confusionM.lReg<-confusionMatrix(predicted.lReg, test.data$Kyphosis)  
print(confusionM.lReg)

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction absent present  
## absent 16 3  
## present 0 1  
##   
## Accuracy : 0.85   
## 95% CI : (0.6211, 0.9679)  
## No Information Rate : 0.8   
## P-Value [Acc > NIR] : 0.4114   
##   
## Kappa : 0.3478   
##   
## Mcnemar's Test P-Value : 0.2482   
##   
## Sensitivity : 1.0000   
## Specificity : 0.2500   
## Pos Pred Value : 0.8421   
## Neg Pred Value : 1.0000   
## Prevalence : 0.8000   
## Detection Rate : 0.8000   
## Detection Prevalence : 0.9500   
## Balanced Accuracy : 0.6250   
##   
## 'Positive' Class : absent   
##

summary(model.lReg)

## Call:  
## nnet::multinom(formula = Kyphosis ~ ., data = train.data)  
##   
## Coefficients:  
## Values Std. Err.  
## (Intercept) -2.802845551 1.651499825  
## Age 0.008487559 0.006541321  
## Number 0.452436071 0.241935524  
## Start -0.116142793 0.077015622  
##   
## Residual Deviance: 51.44055   
## AIC: 59.44055

print(anova(gam2, gam3, gam4, test = "F"))

## Analysis of Deviance Table  
##   
## Model 1: I(Kyphosis == "present") ~ s(Age, 1) + s(Start, 1) + s(Number,   
## 1)  
## Model 2: I(Kyphosis == "present") ~ s(Age, 2) + s(Start, 2) + s(Number,   
## 2)  
## Model 3: I(Kyphosis == "present") ~ s(Age, 8) + s(Start, 8) + s(Number,   
## 8)  
## Resid. Df Resid. Dev Df Deviance F Pr(>F)   
## 1 77 9.9738   
## 2 74 9.1148 3 0.85894 2.3407 0.08286 .  
## 3 57 6.9723 17 2.14247 1.0303 0.44166   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

## Comparing GAM models using residual deviance, which shows how well the model responds to the predictors  
### GAM 2: 9.9738  
### GAM 3: 9.1148  
### GAM 4: 6.9723  
### Original Model: 51.446  
  
## Comparing Models using AIC, a previous used metric to compare models.  
### GAM 2: 70.2143   
### GAM 3: 68.9197  
### GAM 4: 81.2155  
### Original Model: 59.4405  
  
  
# Discussion:   
## This makes me believe model GAM 3 would be the best fit including the solution from our last homework. This is due the model having both the second lowest AIC and Residual Deviance which makes it the best model overall.

#### Question 2

#### Use data set CanadianWeather from R package “fda”

library(fda); data("CanadianWeather")

## Warning: package 'fda' was built under R version 4.1.2

## Loading required package: fds

## Warning: package 'fds' was built under R version 4.1.2

## Loading required package: rainbow

## Warning: package 'rainbow' was built under R version 4.1.2

## Loading required package: pcaPP

## Warning: package 'pcaPP' was built under R version 4.1.2

## Loading required package: RCurl

##   
## Attaching package: 'RCurl'

## The following object is masked from 'package:tidyr':  
##   
## complete

## Loading required package: deSolve

## Warning: package 'deSolve' was built under R version 4.1.2

##   
## Attaching package: 'fda'

## The following object is masked from 'package:lattice':  
##   
## melanoma

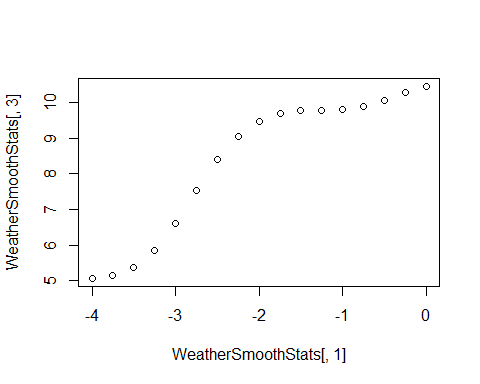
## The following object is masked from 'package:graphics':  
##   
## matplot

#### Question 2.1: Choose the level of smoothing using generalized cross validation criterion to represent CanadianWeather$monthlyPrecip using a functional object.

# Setting up   
Weatherrange <- c(0,365)  
Weatherbasis <- create.fourier.basis(Weatherrange, nbasis=21, period = 365)  
Weathertime <- seq(.5,11.5,1)  
harmaccelLfd <- vec2Lfd(c(0, (2\*pi/365)^2, 0), rangeval=Weatherrange)  
  
  
# Running  
WeatherLoglam <- seq(-4,0,0.25) # Generating the parameter to smooth stats  
nglam <- length(WeatherLoglam)  
  
  
WeatherSmoothStats <- array(NA, dim=c(nglam, 3),  
 dimnames=list(WeatherLoglam, c("log10.lambda", "df", "gcv") ) )  
WeatherSmoothStats[, 1] <- WeatherLoglam  
  
  
for (ilam in 1:nglam) {  
 WeatherSmooth <- smooth.basisPar(Weathertime, CanadianWeather$monthlyPrecip, Weatherbasis,  
 Lfdobj=int2Lfd(2), lambda=10^WeatherLoglam[ilam])  
 WeatherSmoothStats[ilam, "df"] <- WeatherSmooth$df  
 WeatherSmoothStats[ilam, "gcv"] <- sum(WeatherSmooth$gcv)  
  
}  
  
  
  
WeatherSmoothStats

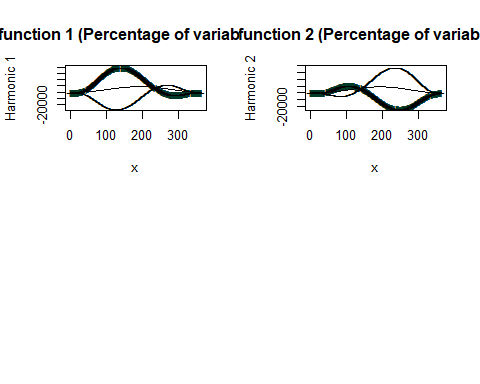
## log10.lambda df gcv  
## -4 -4.00 4.905783 5.070206  
## -3.75 -3.75 4.827190 5.141760  
## -3.5 -3.50 4.720755 5.366030  
## -3.25 -3.25 4.587402 5.850696  
## -3 -3.00 4.440482 6.618788  
## -2.75 -2.75 4.300535 7.537103  
## -2.5 -2.50 4.182929 8.396272  
## -2.25 -2.25 4.090165 9.053270  
## -2 -2.00 4.013829 9.475045  
## -1.75 -1.75 3.940245 9.695406  
## -1.5 -1.50 3.854938 9.772434  
## -1.25 -1.25 3.746599 9.777722  
## -1 -1.00 3.612458 9.794367  
## -0.75 -0.75 3.462505 9.887718  
## -0.5 -0.50 3.315664 10.061247  
## -0.25 -0.25 3.187582 10.262757  
## 0 0.00 3.081255 10.445446

plot(WeatherSmoothStats[, 1], WeatherSmoothStats[, 3])



#### Question 2.2: Perform functional PCA analysis to CanadianWeather$monthlyPrecip. Plot the first two PCs with perturbation. Interpret the first two PCs extracted (what variation pattern each PC represents?).

Weatherfd <- smooth.basisPar(Weathertime, CanadianWeather$monthlyPrecip,  
 Weatherbasis, Lfdobj=harmaccelLfd, lambda=1e-2)$fd  
  
# PCA  
WeatherfdPar <- fdPar(Weatherbasis, harmaccelLfd, lambda=1e-2)  
Weatherpca.fd <- pca.fd(Weatherfd, nharm=2, WeatherfdPar)  
Weatherpca.fd <- varmx.pca.fd(Weatherpca.fd)  
  
  
# Plotting  
op <- par(mfrow=c(2,2))  
plot.pca.fd(Weatherpca.fd, cycle=TRUE)  
par(op)



## Interpret the first two PCs extracted  
# Both functions are plotted with x being days out of the year and y being the variation, Based off these graphs it can be interpreted that the PCA function 1 has very variability at the beginning of the year and the rest rest of the year has very low variation. PCA function 2 is vice-versa. This shows that the two PCA functions are enough to represent the Canadian Weather data because the high variation does not overlap.

#### Question 2.3: (Bonus 2 point) Apply bivariate functional PCA to monthlyPrecip and monthlyTemp. Plot the first two PCs with perturbation (temp vs. precip). Interpret the first two PCs extracted (what variation pattern each PC represents?)

Did not do the bonus

Q3: Use google scholar to find one paper that uses functional data analysis methods to solve the research problem (this cannot be the paper we discussed in class).

Paper Discussed: Classification using functional data analysis for temporal gene expression data.

The research objective was to use machine learning techniques to split each gene expression into gene groups. The researchers proposed a new method which was using functional logistic regression with the implementation of functional principal component analysis. Functional data analysis is very helpful in this dataset since the data collected is sparse. The eigenfunctions of the functional logistic regression was chosen through leave-one-out cross validation. To compare the results this method was compared against the B-Spline implementation of functional data analysis, which is through my knowledge is the industry standard in this field.

After running simulations and the actual experiment the paper had some major contributions. Implementing the proposed methodology successfully and finding great results compared to the industry standards. One of the results was the ability to properly classify a gene sub-group. This has not been done before this paper and is considered a major success. Since the amount of eigenfunctions and base functions are also reduced it is less computationally expensive as well.

Though the comparative results to the industry standard are very good, and it is less computationally expensive as the industry standard there are still limitations to the proposed method. It is still computationally taxing and requires a long time to run. This is mostly due to the leave-one-out cross validation that selects the eigenfunctions. For possible future directions there are also a lot of room for growth since temporal gene expression analysis is a complicated dataset and has a lot of room for improvement in terms of high variation and low accuracy (90%).

Xiaoyan Leng, Hans-Georg Müller, Classification using functional data analysis for temporal gene expression data, Bioinformatics, Volume 22, Issue 1, 1 January 2006, Pages 68–76, https://doi.org/10.1093/bioinformatics/bti742