Lab 6

Table of contents

# Lab 6

In this class, we will expand on the reconstruction error that we briefly touched on last week.

More generally, PCA is a type of unsupervised learning method used for dimension reduction. What do we mean by unsupervised learning? Let’s try and answer this by looking at an past exam question from SP8 (General Insurance Pricing) in September 2023.

**Q1: Explain, in your own words, the differences between supervised and unsupervised learning, in the context of building a predictive model. [5]**

**Solution:**

## W8 Slides I

### Autoencoders - mathematical formulation

Recall from the lecture that an auto-encoder is an unsupervised learning method used for dimension reduction which consists of two mappings:

with dimension . Therefore, an autoencoder typically leads to a loss of information.

We choose a dissimilarity, or distance, function such that if and only if .

Then, an autoencoder is a pair of mappings such that their composition leads to small reconstruction error with respect to the dissimilarity function :

where is the original data and is its projection.

### Autoencoders - PCA as an autoencoder

* The mapping is called encoder, the mapping is called decoder (the encoder compresses the input and the decoder attempts to recreate the input from the compressed version provided by the encoder), and is the -dimensional representation of which contains all information of up to a small reconstruction error.
* The PCA provides a first example of an autoencoder.
* The encoder in the case of PCA is the linear transform of the design matrix by the first columns of the eigenvector matrix and
* The decoder in the case of PCA is the way we transform back to matrix , i.e.,
* The dissimilarity function in the case of PCA is the squared Frobenius norm:

### Autoencoders - reconstruction of the original variables in the motivating data set using PCA

* In the previous lecture, for our motivating data set we chose ( p = 2 ) PCs and calculated the reconstruction error.
* Now we will graphically illustrate all the reconstructed values ( X\_p ) on the y-axis against the original values ( X ) on the x-axis.

library(tidyverse)

Warning: package 'tidyverse' was built under R version 4.2.3

Warning: package 'ggplot2' was built under R version 4.2.3

Warning: package 'tibble' was built under R version 4.2.3

Warning: package 'tidyr' was built under R version 4.2.3

Warning: package 'readr' was built under R version 4.2.3

Warning: package 'purrr' was built under R version 4.2.3

Warning: package 'dplyr' was built under R version 4.2.3

Warning: package 'stringr' was built under R version 4.2.3

Warning: package 'forcats' was built under R version 4.2.3

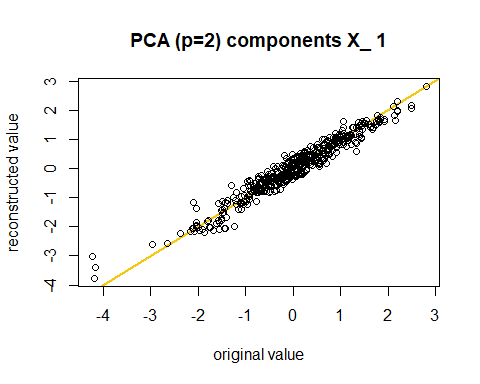
Warning: package 'lubridate' was built under R version 4.2.3

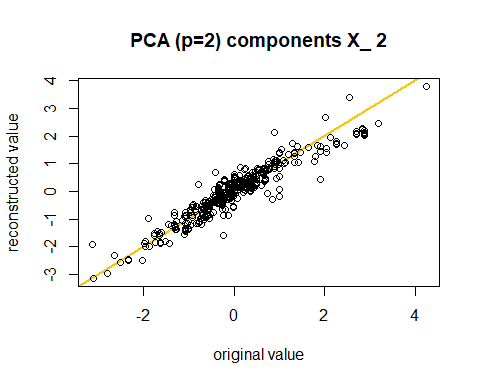
── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──  
✔ dplyr 1.1.4 ✔ readr 2.1.4  
✔ forcats 1.0.0 ✔ stringr 1.5.1  
✔ ggplot2 3.5.1 ✔ tibble 3.2.1  
✔ lubridate 1.9.3 ✔ tidyr 1.3.1  
✔ purrr 1.0.2   
── Conflicts ────────────────────────────────────────── tidyverse\_conflicts() ──  
✖ dplyr::filter() masks stats::filter()  
✖ dplyr::lag() masks stats::lag()  
ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors

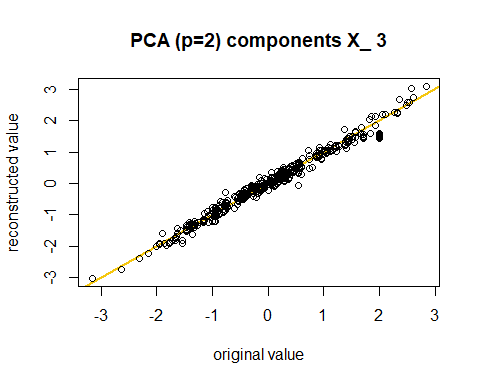
library(ggplot2)  
library(GGally)

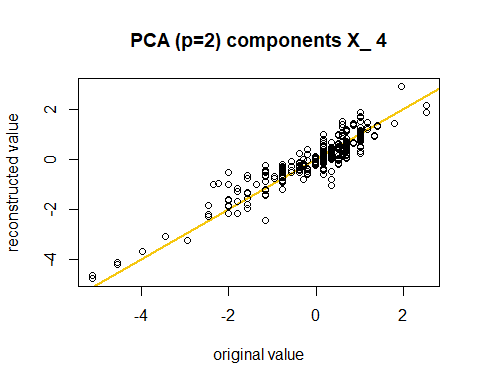
Registered S3 method overwritten by 'GGally':  
 method from   
 +.gg ggplot2

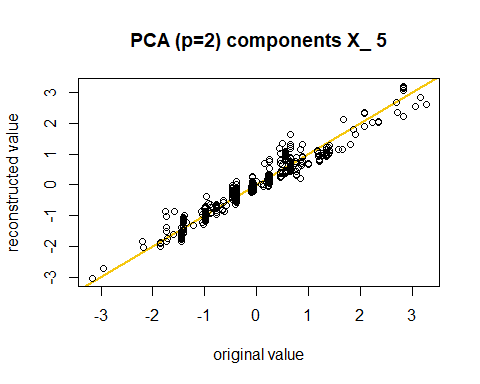
# Step 1: First run the codes: "R Code Lecture Slides Part III",   
# "R Code Lecture Slides Part IV A" and   
# "R Code Lecture Slides Part IV B" or Class 5 (Solution).   
SC <- read.csv('SportsCars.csv', sep = ';', header = TRUE)  
SC <- SC %>% mutate(W\_log = log(weight),  
MP\_log <- log(max\_power),  
CC\_log <- log(cubic\_capacity),  
MT\_log <- log(max\_torque),  
MES\_log <- log(max\_engine\_speed),  
S100\_log <- log(seconds\_to\_100),  
TS\_log <- log(top\_speed))  
S100\_log <- log(SC$seconds\_to\_100)  
SC <- SC %>% mutate(sports\_type = cut(tau, breaks = c(0, 17, 21, 100),   
 labels = c('tau<17 (sports car)',   
 '17<=tau<21', 'tau>=21')))  
SC <- SC %>% mutate(x1 = log(weight/max\_power),  
 x2 = log(max\_power/cubic\_capacity),  
 x3 = log(max\_torque),  
 x4 = log(max\_engine\_speed),  
 x5 = log(cubic\_capacity))  
X\_raw <- SC %>% dplyr::select(x1,x2,x3,x4,x5)  
X <- apply(X\_raw, 2, function(x) (x - mean(x))/sd(x))  
svdX <- svd(X)  
V <- svdX$v  
  
  
# Step 2: Then, compute the reconstructed values in Xp   
# from lecture slides Part IV B, pages 6 and 23:   
Xp <- X %\*% V[,1:2] %\*% t(V[,1:2])   
# Step 3: Plot the reconstructed values Xp against the original   
# values X, where j=1,2,3,4,5 is the corresponding   
# column of X and Xp. # Here we assume j=1, but you should run the   
# code again for j=2,3,4,5:   
for(j in 1:5) {  
 plot(X[,j], Xp[,j],  
 xlab = "original value", ylab = "reconstructed value",  
 main = paste("PCA (p=2) components X\_", j),  
 abline(a=0, b=1, col=7, lwd=2))  
 # Optional: pause between plots so you can see each one  
 Sys.sleep(1)  
}











* If the reconstruction was perfect, then all these points would lie on the orange diagonal line.
* We observe that the reconstruction with PCs works rather well, the most difficult component seems which is the most non-Gaussian one.
* The PCA is optimal if we consider linear approximations under the squared Euclidean distance i.e. $ d(x, y) = ||x - y||^2$
* However, this is not appropriate for .
* Next we will construct **nonlinear autoencoder** using a particular neural network architecture capable of discovering structure within data, the so called **bottleneck neural network** for dimension reduction.
* But before this we will start with a “gentle” introduction to neural networks.

## W8 Slides II

### The Poisson Distribution

The probability mass function (pmf) of Poisson distribution is

with mean

and variance

* It should be noted that the mean of the Poisson distribution is the same as its variance.
* However, in reality, the claims are usually overdispersed due to unobserved heterogeneity. Thus, a model such as the Negative Binomial, whose variance exceeds its mean, is more appropriate for the number of claims.

### The Poisson regression model for predicting insurance claims

* Assume that the number of claims, is denoted by , for policyholder , are independent follow the Poisson distribution

where the parameter depends on the policyholder’s characteristics and an offset of the claims .

* For the Poisson regression with the canonical log-link function, we have that the average number of claims is given by

where is the vector of unknown regression coefficients to be estimated by maximum likelihood estimation (MLE).

Q: What do you think the offset means?

A: It is the exposure, typically the logarithm of the exposure like time, area or population.

### Description of the motivating data set

We used real claim frequency data from a French Motor Third-Part Liability (MTPL) data set in the R package CASdatasets.

* Response is the number of claims, with Exposure as the offset.

The explanatory variables are:

* **Area**: Factor w/ 6 levels “A”, “B”, “C”, “D”, …: 4 4 2 2 2 5 5 3 3 2 …
* **VehPower**: int 5 5 6 7 7 6 6 7 7 …
* **VehAge**: int 0 0 2 0 0 2 2 0 0 …
* **DrivAge**: int 55 55 52 46 46 38 38 33 33 41 …
* **BonusMalus**: int 50 50 50 50 50 50 50 50 68 68 50 …
* **VehBrand**: Factor w/ 11 levels “B1”, “B10”, “B11”, …: 4 4 4 4 4 4 4 4 4 4
* **VehGas**: Factor w/ 2 levels “Diesel”, “Regular”: 2 2 1 1 1 2 2 1 1 1 …
* **Density**: int 1217 1217 54 76 76 3003 3003 137 137 60 …
* **Region**: Factor w/ 22 levels “R11”, “R21”, “R22”, …: 18 18 3 15 15 8 8 20 20 12 …

### Learning set and test set

# Set working directory  
dat2 <- read\_csv("freMTPL2freq.csv")

Rows: 678013 Columns: 12  
── Column specification ────────────────────────────────────────────────────────  
Delimiter: ","  
chr (4): Area, VehBrand, VehGas, Region  
dbl (8): IDpol, ClaimNb, Exposure, VehPower, VehAge, DrivAge, BonusMalus, De...  
  
ℹ Use `spec()` to retrieve the full column specification for this data.  
ℹ Specify the column types or set `show\_col\_types = FALSE` to quiet this message.

# set the random seed  
set.seed(100)  
  
# 90% as the learning data set  
ll <- sample(c(1:nrow(dat2)), round(0.9\*nrow(dat2)), replace = FALSE)  
  
# learning data set  
learn <- dat2[ll,]  
  
# testing data set  
test <- dat2[-ll,]

### Poisson regression for the motivating data set

* We import and preprocess risk features and claim frequency data from French MTPL data set.
* The Poisson regression model is fitted on the claim frequency and 9 risk features, with the exposure as the offset.

Pois.glm <- glm(ClaimNb ~ offset(log(Exposure)) + DrivAge + BonusMalus + VehBrand + Region + VehGas + VehAge + VehPower + Area + Density,   
 data = learn,  
 family = poisson(link = "log"))  
  
learn$Pois.GLM <- fitted(Pois.glm)  
test$Pois.GLM <- predict(Pois.glm, newdata = test, type="response")

### Deviance loss for the Poisson Regression

* To make the regression more in line with the NN model that we will use later for the same purpose, the **deviance loss** is introduced since minimization of deviance loss is equivalent to the process of training NNs.
* The deviance loss is defined as follows:

where is the pmf of the Poisson regression model.

dev.loss <- function(y, mu, density.func) {  
 logL.tilde <- log(density.func(y, y))  
 logL.hat <- log(density.func(y, mu))  
 2 \* mean(logL.tilde - logL.hat)  
}

* In particular, using the Equation for the Poisson model, the **deviance loss** is given by:
* By using the built-in density function for the Poisson distribution:

dev.loss(y = learn$ClaimNb, mu = learn$Pois.GLM, density.func = dpois)

[1] 0.3212739

dev.loss(y = test$ClaimNb, mu = test$Pois.GLM, density.func = dpois)

[1] 0.3134761

### SP8 Exam Questions (Sep 2023)

1. **State the qualities of a good rating factor.** [3]

**A general insurance company that writes a variety of lines of business has been approached by a third party offering data that could be useful as new rating factors.**

1. **Outline the analyses the pricing actuary could undertake to evaluate the potential new factors.** [3]
2. **Suggest considerations that would have to be taken into account before using the factors if the analyses showed they were predictive.** [4]

[Total 10]

|  |
| --- |
|  |