Mini project-1: PCA, biased estimators and their applications

Personal Distribution

Weiqi Xiao (20374988): Be responsible for Python coding and writing; mainly analyze handwritten digits' dataset.

Chang Zhu (20377162): Be responsible for R Coding and writing; mainly analyze animal sleep dataset.

Chengyuan Zhou (20381515): Be responsible for research and writing; mainly research on different linear regression models and package.

Introduction

In this report, we discuss PCA and biased estimators in different part (A and B) along with analysis in different datasets. Below are our analysis reports.

A. PCA and its application

1. Data-handwritten digits

Sources: http://statweb.stanford.edu/~tibs/ElemStatLearn/datasets

Data description: normalized handwritten digits, automatically scanned from envelopes by the U.S. Postal Service. The original scanned digits are binary and of different sizes and orientations; the images here have been descanted and size normalized, resulting in 16 x 16 grayscale images (Le Cun et al., 1990).

2. Designed question

How to catch the main characteristics of different digits? What methods can we use to figure this out? How do we evaluate this method?

3. Analysis procedure

We try to use PCA method to find out the main characteristics of digits. Sine there are over 500 observations of each digits presenting as follows:

0	1	2	3	4	5	6	7	8	9	Total
1194	1005	731	658	652	556	664	645	542	644	7291

By using PCA, the number of components is less than or equal to the smaller of (number of original variables or number of observations). This transformation is defined in such a way that the first principal component has the largest possible variance (that is, accounts for as much of the variability in the data as possible). So, we can get the components that can explain most of the data. Here we use python to analysis data. We are going to do with 10 digits separately and we try to limit the primary components up to be 50. Source code is presented in the Appendix Ⅰ (Just one digit is demonstrated in the code. We just need to change the data source to deal with other digits).

4. Results

Here we take digit 0 for demonstration. First, we draw the first hand-written digit 0. Secondly, we draw the mean of the 1194 observation of digit 0. See figure 1, 2.

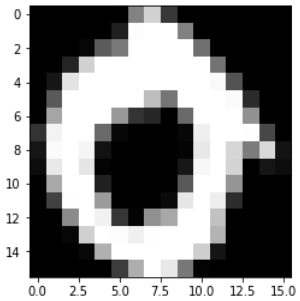


Figure 1

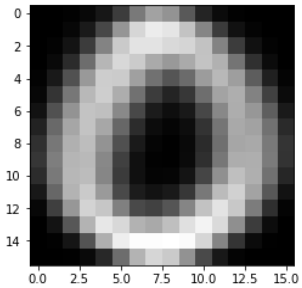


Figure 2

Then we use PCA to analyse this data. See figure 3 and figure 4 which represent the explained variance ratio of each component. From figure 3 and 4, we can see that the top 10 components explain 45.7% of the whole data and the top 50 components explain 65.4%. It means that we can use 50 primary components to explain main characteristics of digit 0, which is less than 1100 more observations. And we draw the top 25 components of digit 0 as figure 5. As the primary components are displayed, we can use the primary components to do other model like knn and SVM etc. to bulid up models to distinguish digits. But here we do not make any further model to do classicification. We also to PCA with other 9 digits and we find out that they are all similar to digit 0 that we do at this report. See figures of other digits analysis in Appendix II .We can see that the top 17 components only explain 80.8% of the whole observation and 93.6% explained variance from 50 primary components. It can explain much of the whole data. What is more, the components after 10^{th} components explain very few of the observation. So whether choose more than 10 components or 50 compenents is a big problem for us. Actually we will choose 50 components to do further models since we reduce a lot from 1100 more observations to only 50 components.

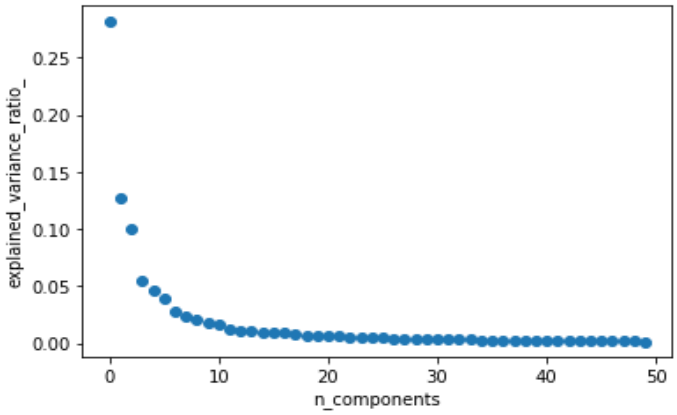


Figure 3

```
In [13]: sum(pca.explained_variance_ratio_[0:15])
Out[13]: 0.79872263154536116

In [14]: sum(pca.explained_variance_ratio_[0:16])
Out[14]: 0.80823492012754505

In [15]: sum(pca.explained_variance_ratio_[0:50])
Out[15]: 0.93560805107421918
```

Figure 4

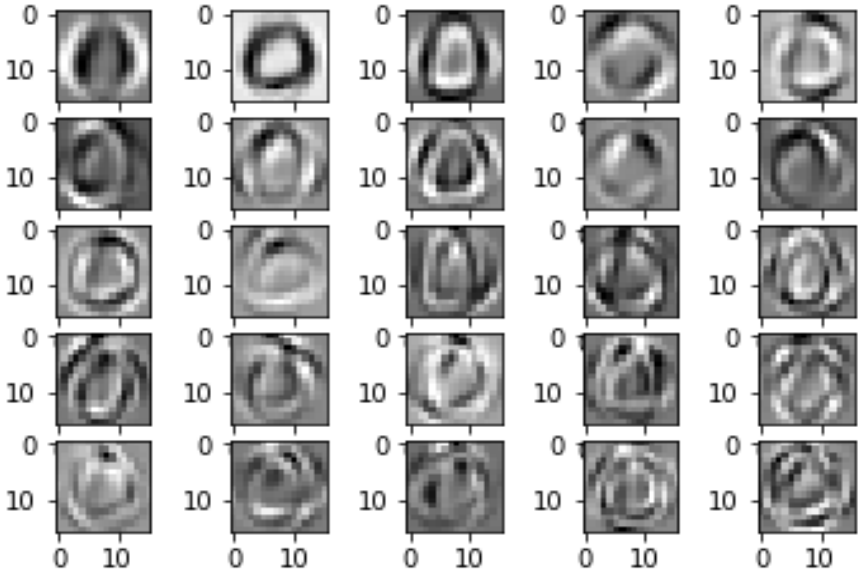


Figure 5

B. Biased estimators and its application

1. Data-animal sleeping data

Sources: http://math.stanford.edu/~yuany/course/data/sleep1.csv

Data description: There are 62 species.

Variables (column):

-Species

-Body Weight (kg)

-Brain Weight (grams)

-Non Dreaming Sleep (hours)

-Dreaming Sleep (hours)

-Total Sleep (hours)

-Maximum Life Span (years)

-Gestation Time (days)

-Predation Index (1=Least Likely, 2=Not Likely, 3=Neutral, 4=Likely, 5=Most Likely)

-Sleep Exposure (1=Highly Unexposed, 2=Unexposed, 3=Neutral, 4=Exposed, 5=Highly Exposed)

-Dangerous (1=Least Dangerous, 2=Not Dangerous, 3=Neutral, 4=Dangerous, 5=Most Dangerous)

2. Designed question

What is the relationship between sleep hours and other features? What methods can we use to determine the relationship? How can we explain the results after data analyzing?

3. Analysis procedure

We try to do linear regression to look for relationship between sleep hours and other features. At the same time, a problem come along with this idea. How can we estimate the coefficient of other features. What we learnt in class is to use LASSO or Linearized Bregman Iteration to estimate the coefficient. What we want to do is to make a comparison between LASSO and Linearized Bregman Iteration and we choose LASSO to select variables which are significant to sleep hours.

4. Results

The first thing we do is to deal with the NA values in the dataset. Since we do not want to delete the data, we decide to use knn imputation method to get rid of the NA values. And then we have to standarize the input data to avoid big magnitude effect of variables. Here we use R to build the model and find the source R code in Appendix Ⅰ.

As figure 6 shows when sleep is dependent variable, LASSO path is continuous while ISS path is piece-wise. But Linearized Bregman iterations lie between them. We also know that when kappa goes to infinit, it becomes a path like ISS.

By using lasso, we can see that brain, predation, sleepExposure and danger always have larger (greater than zero) or smaller (less than zero) coefficient. Also, we use ISS solver for linear model with lasso penalty to estimate and we can get the similar result.

At the next step, we try to use LASSO to select variables which are significant to sleep/slowWaveSleep/dreamSleep. What we use as an selection standard is Mallows's $\mathrm{C_P}$ value. A small value of $\mathrm{C_p}$ means that the model is relatively precise. We choose the variables at the step where its $\mathrm{C_p}$ value is the minimun.

Maiilos's $\mathrm{C_p}$ is used to assess the fit of a regression model. It is applied in the context of model selection. Where a number of prediction variables are available for predicting some outcome, and the goal is to find the best model involving a subset of these predictors. A small value of $\mathrm{C_p}$ means that the model is relatively precise.

Figure 7/8/9 shows when slowWaveSleep/dreamSleep/sleep is dependent variable, we try to select the variables at the 6th step taking dreamSleep for an example. It can be seen that C_p value is 2.7665 which is the minimun at the step 6. So we choose life, gestation, predation, danger as significant variables. It is the same to select variables in other models.

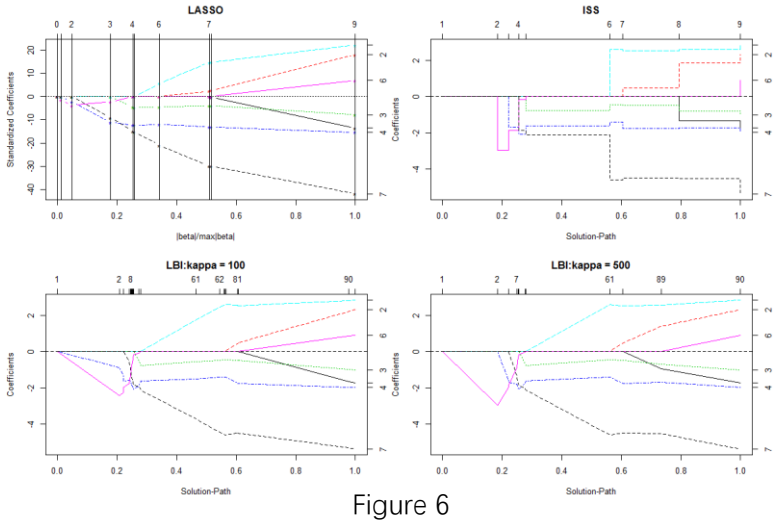
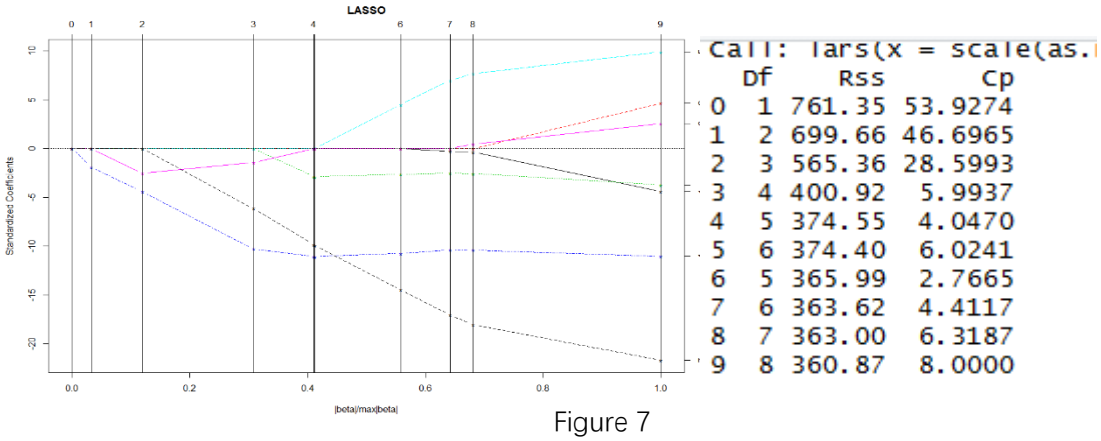


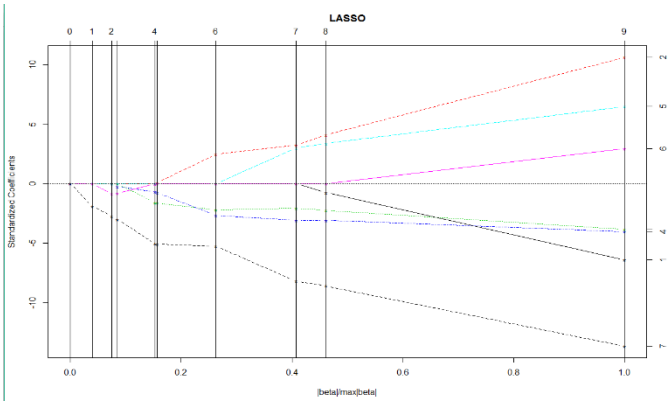
Figure 6



Call: lars(x = scale(as.

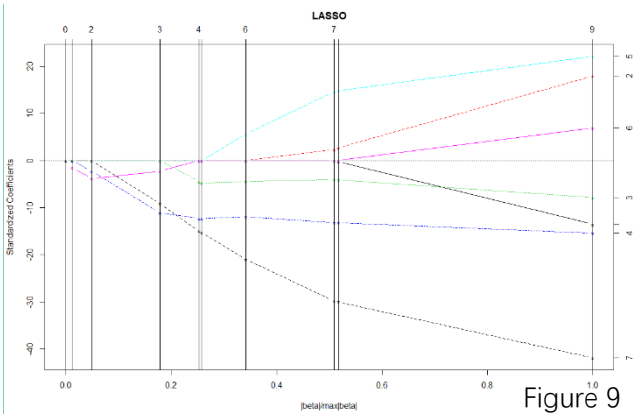
	Df	Rss	Cp
0	1	761.35	53.9274
1	2	699.66	46.6965
2	3	565.36	28.5993
3	4	400.92	5.9937
4	5	374.55	4.0470
5	6	374.40	6.0241
6	5	365.99	2.7665
7	6	363.62	4.4117
8	7	363.00	6.3187
9	8	360.87	8.0000

Figure 7



	Df	Rss	Cp
0	1	118.292	62.6336
1	2	97.353	42.9265
2	3	84.286	31.3800
3	4	81.600	30.5950
4	5	69.076	19.6112
5	6	68.754	21.2780
6	5	61.395	11.6490
7	6	57.183	9.2823
8	7	56.269	10.3346
9	8	52.088	8.0000

Figure 8



	Df	Rss	Cp
0	1	1286.74	85.9982
1	2	1219.51	80.3703
2	3	1035.68	61.5116
3	4	638.23	18.4156
4	5	559.70	11.5051
5	4	557.75	9.2840
6	5	530.55	8.1981
7	6	499.68	6.6953
8	7	498.94	8.6111
9	8	475.92	8.0000

Figure 9

Appendix I

1. Source code for PCA-python

```
import pandas as pd
import io
import requests

import numpy as np

####load data from train0-9
url="http://statweb.stanford.edu/~tibs/ElemStatLearn/datasets/zip.digits/train.0"
s = requests.get(url).content
c = pd.read_csv(io.StringIO(s.decode('utf-8')))
data = np.array(c,dtype='float32');
data.shape

# Reshape the data into image of 16x16 and show the image.
import matplotlib.pyplot as plt

img1 = np.reshape(data[1,:],(16,16));
imgshow = plt.imshow(img1,cmap='gray')



# Now show the mean image.
mu = np.mean(data, axis=0);
img_mu = np.reshape(mu,(16,16));
imgshow = plt.imshow(img_mu,cmap='gray')

##########################################
# PCA

from sklearn.decomposition import PCA
pca = PCA(n_components=50)
pca.fit(data)

print(pca.explained_variance_ratio_)

# Plot the 'explained_variance_ratio_'

plt.plot(pca.explained_variance_ratio_, "o", linewidth=2)
plt.axis('tight')
plt.xlabel('n_components')
plt.ylabel('explained_variance_ratio_')
```

```
# Principal components

Y = pca.components_;
Y.shape
# Show the image of the top 25 principal components

for i in range(25):
        plt.subplot(5,5,i+1)
        img_pca = np.reshape(Y[i,:],(16,16))
        imgshow = plt.imshow(img_pca,cmap='gray');
```

2. Source code for biased estimators-R

```
#####animal sleep-NA vlues
sleep <- read.csv("~/Desktop/6380J/animal sleep.csv")
a <- sleep[!is.na(sleep)]
library(DMwR)
sleepnew <- sleep
sleepnew <- knnImputation(sleepnew)
row.names(sleepnew) <- sleepnew$species
###########LAsso iss libra
library(lars)
library(Libra)

lasso <- lars(scale(as.matrix(sleepnew[,-c(1:3)])),sleepnew$slowWaveSleep)
par(mfrow=c(2,2))
plot(lasso)
lasso$Cp[which.min(lasso$Cp)]
issobject <- iss(scale(as.matrix(sleepnew[,-c(1:3)])),sleepnew$sleep)
plot(issobject,xtype="norm")    #plot.lb
title("ISS",line = 2.5)

kappa <- c(100,500)
for (i in 1:2){
   object<-lb(scale(as.matrix(sleepnew[,
c(1:3)])),sleepnew$dreamSleep,kappa[i],family="gaussian",trate=100)
   plot(object,xtype="norm")
   title(paste("LBI:kappa =",kappa[i]),line = 2.5)
}
detach(lasso)
summary(lasso)
###variables selection
lasso1 <- lars(scale(as.matrix(sleepnew[,-c(1:3)])),sleepnew$slowWaveSleep)
lasso1$Cp[which.min(lasso1$Cp)]
summary(lasso1)
```

```
plot(lasso1)

sleep.lm <- lm(sleepnew$slowWaveSleep~scale(as.matrix(sleepnew[,-c(1:3)])),)
summary(sleep.lm)
```

Appendix II

1. Figures of other digits

The results of other digits(1 to 9) are very similar with digit 0 so we just display some figures of the results which are the top 16 primary components: