Regression Analysis: *Revised* HW 3

Matthew Tichenor

September 28, 2015

1 Introduction

Principle Components Analysis is a popular technique used in data reduction. Essentially, principle components analysis is a way of visualizing an n-dimensional data set in fewer dimensions, by using the principal directions, which account for most of the variation, as a basis. For an nxp matrix X the singular value decomposition is given by $X = U \cdot L^{\frac{1}{2}} \cdot Z'$, where U contains the eigenvectors of $X \cdot X'$, Z contains the eigenvectors of $X' \cdot X$ and $L^{\frac{1}{2}}$ contains the square root of the associated eigenvalues. Using this decomposition, let $W = X \cdot Z = U \cdot Z^{\frac{1}{2}}$, then the columns of W, called the principle component scores, are a linear transformation of the columns of X by the eigenvectors in Z. More importantly, the sum of squares of the columns of W equal the eigenvalue associated with the column of Z in the transformation. The sum of eigenvalues equals the total sum of squares, and in this way, by using only a few of the principle components, a large proportion of the total sum of squares (or variance) can be captured.

These ideas presented here were tested on two different datasets: the swiss dataset, and a dataset of images given by the instructor, Dr. Su. In the next sections, principle components analysis are applied to those datasets.

2 PCA on the Swiss Dataset

The swiss dataset is an example that comes with R. The dataset contains demographic measures of 47 provinces of Switzerland during the year 1888. The data set contains 47 observations (one for each province) and 6 variables: fertility, agriculture, examination, education, Catholic and infant mortality. The fertility variable is a standardized fertility measure. The agriculture variable is a measure of the percentage of males in agricultural occupations. The examination variable is the percentage of army recruits with highest marks on their exams. The education variable is the percentage of army recruits who have education beyond primary school. The Catholic variable is the percentage of people who are Catholic, as opposed to some other version of Christianity. The infant mortality rate is the percentage of children who die within 1 year.

It is important to note the total percentage of the variance explained by the principal components. In the table below, the cumulative variance for first 3 principal components are given. The first 2 principal directions capture 73% of the total variation in the data set. This means, most of the variation for this data set can be viewed in 2-d. If 73% is not good enough for you, then the first 3 principal directions capture all but 13% of the total variation. Adding a fourth principal direction captures 95% of the total variation.

Table 1: Cumulative Variance

PC1	PC2	PC3	PC4	...
0.533	0.7313	0.8726	0.946	...

One way to visualize this is by using a scree-plot. See Figure 1d. Each point on this plot is $\sum_{j=1}^{i} \lambda_j / \sum_{j=1}^{n} \lambda_j$, so this plot can be used to see the relative weight each eigenvalue has on the total sum of squares.

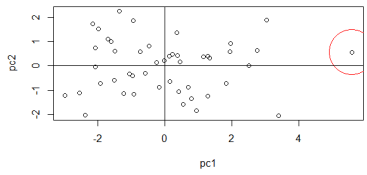
The first 3 principal directions (or rotations) of this dataset are summarized below.

Table 2: The Principal Directions

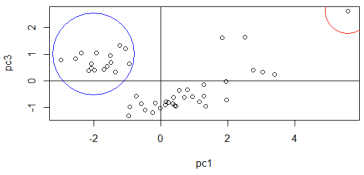
Var	PC1	PC2	PC3
Fertility	-0.457	0.322	-0.174
Agriculture	-0.424	-0.412	0.038
Examination	0.510	0.125	-0.091
Education	0.454	0.179	0.532
Catholic	-0.350	0.146	0.807
Inf Mort	-0.150	0.811	-0.160

We see PC1 is a linear combination of the 6 variables with Catholic and Infant Mortality having the lowest weights. PC2 is a linear combination of the 6 variables with Examination and Catholic having the lowest weights. Since Catholic has very little weight on PC1 and PC2, this variable could be ignored. If the first 3 principal components were used, then we would not ignore Catholic, since this variable has a significant weight on PC3.

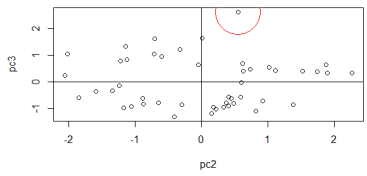
In Figures 1a, 1b and 1c, the points of the data set (provinces) are plotted along the axes given by the principle directions [1]. These graphs are only ways of visualizing the data in 2 dimensions. The data has not been altered in any way.



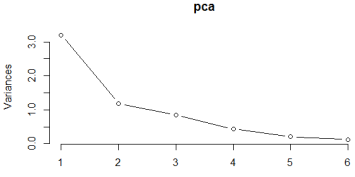
(a) PCP 1 & 2



(b) PCP 1 & 3



(c) PCP 2 & 3



(d) Scree Plot

[1] While it is possible to label the points on these graphs, the majority of these labels would overlap and make the points indistinguishable. For clarity and aesthetic purposes, labels weren't given for these graphs

Principal Components Analysis may be used in detecting outliers. Examining Figure 1a more closely, one observation is farther from the others. This observation, the province V. De Geneve, is circled in red. This same observation is circled in both Figures 1b and 1c, where in Figure 1b the observation is clearly separate from other observations. Thus the Swiss Province V. De Geneve is an extreme observation[2], and should be investigated further for explanation.

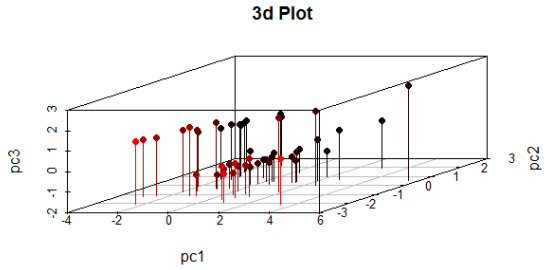


Figure 2: PC1 vs PC2 vs PC3

In addition, one group of observations seems to be distinct from the rest in Figure 1b. These observations are circled in blue. This grouping of observations does not appear in Figures 1a and 1c, so it is not clear, using only 2-d plots, whether or not these observations form a distinct group. Figure 2 is a 3-d plot along the first 3 principal directions, where it is clear that some of these observations maybe treated as separate groups[3].

3 PCA on Images

Here the dataset are images. Two hundred 28x23 pixel, grayscale images were concatenated into row vectors, then concatenated to form a 200x644 matrix. Each row of the matrix was an image, while each column represented one of the 644 pixels, which were treated as variables. Figure 3 shows all 200 images. Each image is a picture of someone's face.

[2]This interpretation is subjective.
[3]This interpretation is also subjective

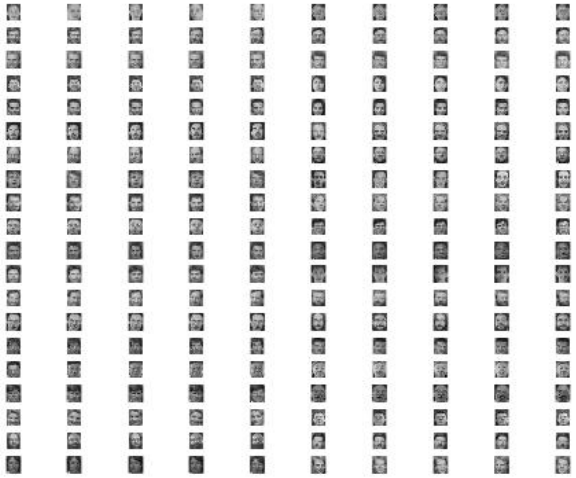
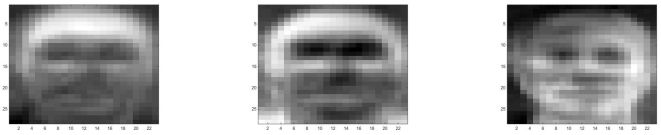


Figure 3: 200 Images

After centering the data, the eigenvectors for the covariance matrix were found. The eigenvalues are the principal directions in principal component analysis. The corresponding images for the first 3 principal directions are shown in Figures 4a, 4b and 4c. As gray-scale images these vectors look like faces. Principal direction 1 is the eigenvector associated with the largest eigenvalue. Principal direction 2 is the eigenvector associated with the second largest eigenvalue and so forth. The eigenvectors combined make up the Image Space, and the eigenvectors are called "Eigen-Faces".

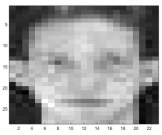


(a) Direction 1 (b) Direction 2 (c) Direction 3

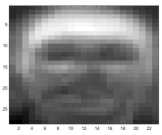
An interesting thing happens when an image is projected onto a subspace spanned by the principal directions. The eigenvectors of the covariance matrix are orthogonal. Because of this, the projection onto the subspace spanned by these eigenvectors is the sum of the projections onto each vector, that is $p(y|V) = \sum_{i=1}^{k} p(y|x_i)$, where $p(y|x_i) = \frac{<y,x_i>}{||x_i||^2} \cdot x_i$. The eigenvectors returned in MATLAB functions are chosen to have unit length, so $p(y|V) = \sum_{i=1}^{k} < y, x_i > \cdot x_i$.

Figure 5a is the first image of the dataset. Figure 5b is the projection of this image onto the subspace spanned only by the first principal direction. It is a projection onto a 1 dimensional subspace. A face is visible, although it does

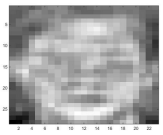
not resemble the original image (or anything remotely human). Figure 5c is the projection of the first image onto the subspace spanned by the first 21 principal directions. This projection resembles something more akin to the original image. Subsequent projections are images that look more and more similar to the original image with the addition of principal directions. Figure 5d is the projection onto the first 31 principal directions, and Figure 5e is the projection onto the first 101 principal directions. Projecting the image onto all principal directions is a projection onto the image space itself, which gives you the original image as demonstrated in Figure 5f.



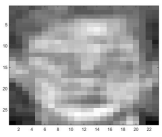
(a) Original



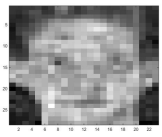
(b) Onto E.V.1



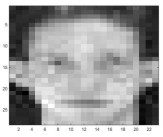
(c) Onto E.V.1-21



(d) Onto E.V.1-31



(e) Onto E.V.1-101



(f) Onto E.V.1-644

4 code

Both R and MATLAB were used in this assignment. The following subsections contain the lines of code used to complete this homework.

4.1 R code

```
swiss
getwd()
pca = prcomp(swiss,scale.=TRUE);
pc1 = pca$x[,1];
pc2 = pca$x[,2];
pc3 = pca$x[,3];
png=(filename="pc1vspc2.png")
plot(pc1,pc2)
points(x=5.59516797,y=0.55731309,col="red",cex=10)
abline(h=0,v=0)
dev.off()
png=(filename="pc1vspc3.png")
plot(pc1,pc3)
abline(h=0,v=0)
points(x=5.59516797,y=2.62736774,col="red",cex=10)
points(x=-2,y=1,col="blue",cex=18)
```

```
dev.off()
png=(filename="pc2vspc3.png")
plot(pc2,pc3)
abline(h=0,v=0)
points(x=0.55731309,y=2.62736774,col="red",cex=10)
dev.off()
summary(pca)
png=(filename="screeplot.png")
plot(pca,type="l")
dev.off()
pca
install.packages("scatterplot3d")
library(scatterplot3d)
scatterplot3d(pc1,pc2,pc3,pch=16,highlight.3d=TRUE, type="h", main="3d Plot")
savehistory(file = "Homework3.Rhistory")
```

4.2 MATLAB code

```
load('HA4c_dat.mat')
figure(1)
hold on
m = 5; n = 5;
for j = 1:25
img = X(j,:);
img = img';
img = reshape(img,[28,23]);
subplot(m,n,j);
imshow(img,[0,255])
end
figure(1);
clf;
image = X(1,:);
image = image';
image = reshape(image,[28,23]);
imshow(image,[0,255])
figure(1); clf;
a = X(1,:);
a = a';
xbar = mean(X);
for j = 1:200
X(j,:) = X(j,:) - xbar;
end
A = cov(X);
[U,S,V] = svd(A);
ev1 = U(:,1);
ev1 = reshape(ev1,[28,23]);
figure(1);
clf;
```

```
imagesc(ev1)
colormap(gray)
ev2 = U(:,2);
ev2 = reshape(ev2,[28,23]);
imagesc(ev2)
colormap(gray)
ev3 = U(:,3);
ev3 = reshape(ev3,[28,23]);
imagesc(ev3)
colormap(gray)
v = U(:,1);
v = dot(a,v)*v;
v = reshape(v,[28,23]);
imagesc(v)
colormap(gray)
v = zeros(644,1);
for j = 1:21
ev = U(:,j);
ev = dot(a,ev)*ev;
v = v + ev;
end
v = reshape(v,[28,23]);
imagesc(v)
colormap(gray)
v = zeros(644,1);
for j = 1:31
ev = U(:,j);
ev = dot(a,ev)*ev;
v = v + ev;
end
v = reshape(v,[28,23]);
imagesc(v)
colormap(gray)
v = zeros(644,1);
for j = 1:101
ev = U(:,j);
ev = dot(a,ev)*ev;
v = v + ev;
end
v = reshape(v,[28,23]);
imagesc(v)
v = zeros(644,1);
for j = 1:644
ev = U(:,j);
ev = dot(a,ev)*ev;
v = v + ev;
end
v = reshape(v,[28,23]);
imagesc(v)
colormap(gray)
```

```
diary off
```