Homework 2: PCA

Evan Ko

February 26, 2018

Abstract

This paper presents the versatility of the singular value decomposition (SVD) and principal component analysis(PCA). Singular value decomposition is one of the most important and useful factorizations in linear algebra. Noisy data is taken from video cameras pointed at a mass-spring system. Our goal is use Pca to compute the most meaningful basis to re-express a noisy, gar- bled data set. The new basis will get rid of noise and redundancy

1 Introduction and Overview

Three cameras take video of the mass-spring system. From our intuitive knowledge we know that only one camera is needed to capture the movement of the spring in the z direction. But in a naive thinking manor we do not know the actual coordinate system. We will take different measurements from each camera, who has their own coordinate systems. The goal is to get rid of noise and redundancy to do data analysis on the system.

2 Theoretical Background

We first construct our data matrix X from cameras a,b,c

$$X = \begin{bmatrix} x_a \\ y_a \\ x_b \\ y_b \\ x_c \\ y_c \end{bmatrix} \tag{1}$$

The matrix $X \in \mathbb{R}^{mxn}$ where m represents the number of measurement types and n represents the number of data points taken from the camera over time The way to check redundant data is by checking the covariance between data sets. The covariance between two data sets is found by

$$\sigma_{ab}^2 = \frac{1}{n-1} ab^T \tag{2}$$

Since there are many data sets we construct the covariance matrix where X is the data matrix constructed in 1

$$C_x = \frac{1}{n-1} X X^T \tag{3}$$

The diagonals of the covariance matrix represent the variance of the particular measurements. The off diagonals are the covariance between measurements. C_x then reports all the correlations between the data sets.If a diagonal term is large there is a large redundancy while small terms represent statistically independent terms. The ultimate goal is to remove redundancy and identify signals of maximum variance.

We want to diagonalize the covariance matrix to get maximum variance and zero on all the off diagonals. Thus we use the singular variance composition. The SVD will give us an ideal basis in which C_x can be diagonalized such that the redundancies have been removed and the largest variance are ordered. That basis is written in its *principle component.* The SVD is a factorization of any matrix X into a number of constitutive components U,Σ, V where:

$$X = U \Sigma V^* \tag{4}$$

Thus we can define the variable Y = $U^* X$ and the covariance matrix of Y can be found

$$\begin{aligned} C_Y &= \frac{1}{n-1} Y Y^T \\ &= \frac{1}{n-1} (U^* X)(U^* X)^T \\ &= \frac{1}{n-1} U (X X^T) U \\ &= \frac{1}{n-1} U^* U \Sigma^2 U U* \\ &= \frac{1}{n-1} \Sigma^2 \end{aligned} \tag{5}$$

Thus the principle component projection can be crafted using the svd

3 Algorithm Implementation and Development

1. the first order of business was to locate the mass object in the video. The mass in this case was a paint can was found using these techniques:
   - In the video there is a flashlight on top of the paint can. The flashlight was the brightest object in the video
   - Looping through each frame, the image was converted to black and white with a level of 0.95
   - then the coordinates of the maximum value of the matrix was found and the median was found to find an actual pixel

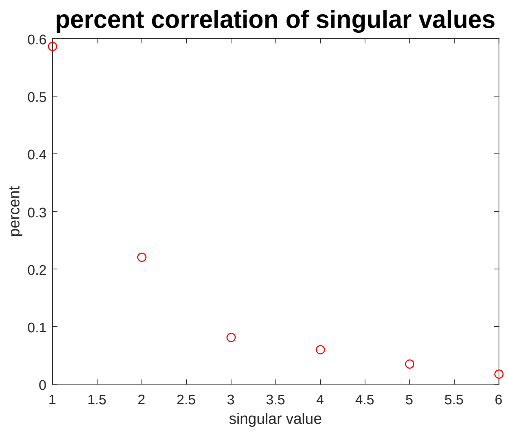
- the object was thus found at the maximum brightness of matrix and thus it was added to the vector of coordinates

2. the data matrix X was constructed as said in equation 1
3. the singular values were plotted as a percentage correlation of singular values
4. then the matrix Y was plotted along with the particular modes labeled to see the most ”heavy”

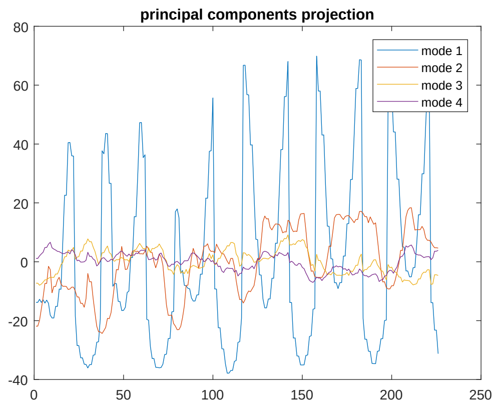
4 Computational Results

4.1 test 1: ideal case

The following data was found from the ideal case
As we can see the first singular value takes about 60% of the weight with the



(a) percent correlation of singular values of test case 1

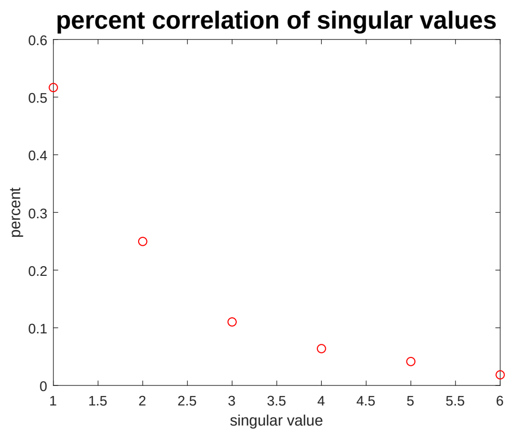


(b) principle components projection of test case 1

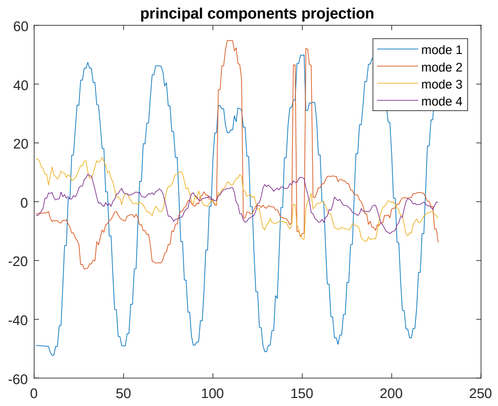
second singular value about 20%. This signifies that there is one direction that is the direction of movement. This lines up with our intuitive knowledge of the problem where the mass moves in only the z direction. As we see in the projection graph mode 1 has the largest amplitude lining up with our intuitive knowledge also.

4.2 test 2: noisy case

The following data was found from the noisy case
As we can see the first singular value takes about 50% of the weight with the second singular value taking about 25%. This signifies that there is still 1 direction that is the direction of movement. We also see that with the addition of



(a) percent correlation of singular values of test case 2



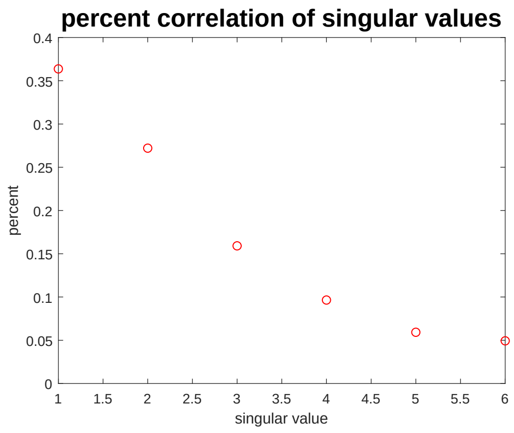
(b) principle components projection of test case 2

noise in the data collected the claim that there is only one direction of movement is not as strong.This lines up with our intuitive knowledge of the problem where the mass moves in only the z direction Since the camera shook there were movements in another direction but small. The noise was filtered out by PCA

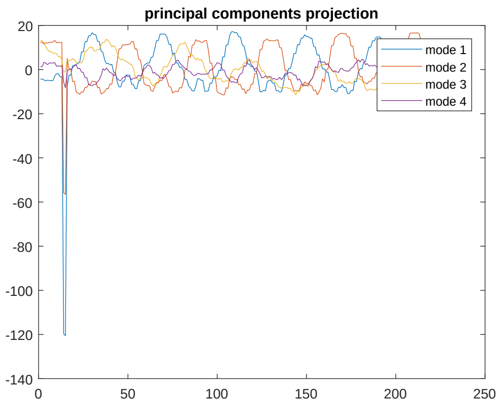
4.3 test 3: horizontal displacement

The following data was found from the ideal case

As we can see the first singular value takes about 35% of the weight with the



(a) percent correlation of singular values of test case 3

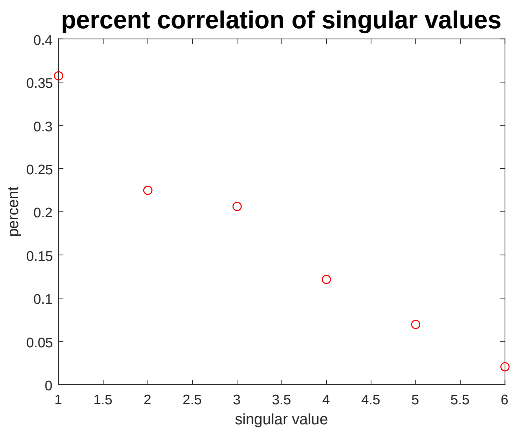


(b) principle components projection of test case 3

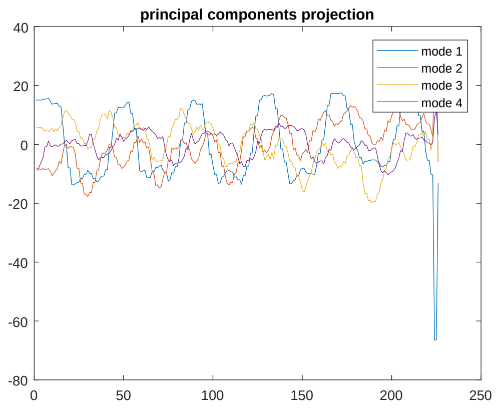
second taking about 27%. This signifies that the paint can is moving in two directions. This lines up with our intuitive knowledge of the problem. The addition of the paint can swinging was caught by PCA and shows up in the data. Interestingly noted in the principle component graph there was a large dip. This can be attributed to noise as the paint can never went down to -120 in any of the other tests. We think that the flash light was hard to pinpoint at the beginning in the video.

4.4 test 4: horizontal displacement and rotation

The following data was found from the ideal case
As we can see the first singular value takes about 35% of the weight with the



(a) percent correlation of singular values of test case 4



(b) principle components projection of test case 4

second taking about 23% and the third mode about 22%. One can guess that from the singular values the paint can was moving in 3 directions but no singular value was taking especially a high percent of correlation. This doesn't line up with our intuitive knowledge of the test case. PCA caught the motion in the x-y plane and z plane but it did not catch the rotation. None of the modes in the principal component projection amplitude was larger than any of the others signifying that there were movements in many directions.

5 Summary and Conclusions

For the most part PCA did a good job finding a perfect basis for each test case. The different motions were captured in the percentage of the singular values. For test case 1 and 2, we knew that there was only motion in the z direction and PCA told us that. In test case 3, we knew that there was only motion in the x-y plane and the z direction and PCA told us that too. In test case 4, there was motion in the x-y plane and the z direction but also rotation. PCA only told us that there were direction in x-y and z direction but not the rotations. One of the reasons for that was the inability to see the flashlight on top of the paint can as it spun away from the cameras. The location of the paint can was then assumed to be another part. I think that was a limitation of my algorithm. I only took physical data to be the location of the paint can but with some tweaking maybe an inferences of the location as the flashlight spun around could be made.

6 Appendix A

- insertMarker: add a marker at specified location, used to show the location of the paint can in the video
- imbinarize: convert the image to binary
- find(I == max(I(:))): used to find the largest value in the matrix I, where the largest value was the brightest

7 Appendix B

hw2script

```
close all; clc;
cropR1 = ["150:375","100:375","100:375"];
cropR2 = ["200:375","100:425","100:425"];
cropR3 = ["225:400","175:375","175:375"];
cropR4 = ["250:400","125:375","100:300"];


cropC1 = ["300:375","250:375","250:500"];
cropC2 = ["300:400","200:400","275:500"];
cropC3 = ["250:400","225:400","250:500"];
cropC4 = ["300:425","200:450","300:500"];
color = ['r','b','g'];
%%
X = zeros(6,226);
for l = 1:3
I1 = eval(strcat('vidFrames',num2str(l),'_1'));
cropI1 = I1(eval(cropR4(l)),eval(cropC4(l)),:,:);
x1 = 1:226;
y1 = 1:226;
for k = 1:226
i = cropI1(:,:,:,k);
I = imbinarize(i); %convert to binary
[r1,c1] = find(I == max(I(:))); %find the coordinates of the brightest spot
%find the average position in the matrix
y1(k) = median(r1);
```

```
x1(k) = median(c1);
end
X(2*l-1,:) = x1;
X(2*l,:) = y1;
end

%% Start analysis of videos
[m,n]=size(X); % compute data size
mn=mean(X,2); % compute mean for each row
X=X-repmat(mn,1,n); % subtract mean

[u,s,v] = svd(X); %get the svd
o = diag(s);
Y = u.'*X; %principal component projection
Cy = (1/(n-1))*Y*Y.'; %covariance matrix of zero redundancy
%%
figure(1);
plot(o/sum(o),'ro'),title('percent correlation of singular values','fontsize',18),xlabel('si

%%
figure(2);
plot(Y(1:4,:).'), title('principal components projection'), legend('mode 1','mode 2','mode 3
```

helping script to find the location of paint can in cropped video frames

```
close all; clc
image = 50;
I1 = vidFrames3_2(:,:,:,:);
I2 = vidFrames3_4(100:300,300:500,:,:);
[a1,b1,e1,d1] = size(I1);
[a2,b2,e2,d2] = size(I2);

%%
%see the full video and the location of the paint can
for k= 1:d1
i1 = I1(:,:,:,k);
g1 = rgb2gray(i1);
[r1,c2] = find(g1 == max(g1(:))); %find the coordinates of the brightest spot
%find the average position in the matrix
x1 = median(c2);
y1 = median(r1);
rgb1 = insertMarker(g1,[x1,y1],'o','color','red','size',10);
imshow(rgb1)
end
```

```
%%
%see the cropped video and the location of the paint can
for q= 1:d2
i2 = I2(:,:,:,q);
g2 = im2bw(i2,0.95);
[r2,c2] = find(g2 == max(g2(:))); %find the coordinates of the brightest spot
%find the average position in the matrix
x2 = median(c2);
y2 = median(r2);
rgb2 = insertMarker(im2uint8(g2),[x2,y2],'o','color','red','size',10);
imshow(rgb2)
end
```