Homework 2: PCA

Yicheng Duan

AMATH 482

02/21/2020

Abstract

In this homework, we three cameras under four different condition shoot videos of a moving mass with a spring. Each camera is in different angle of view and each condition has their own oscillation property. We need extract out coordinate information from three cameras' video in four set of condition. Then, put the coordinates, six principal components from three camera, into a single matrix to perform PCA. Finally, we plot the data in principal components basis to visualize the oscillation of the mass.

Introduction and Overview

First, we have raw data from three cameras. Those data are oversampled since we only want to find out the motion of mass in Z direction. In each of four condition/test, we have six pare of X and Y coordinates of the mass in each frame. For each condition, we put the six pare of X and Y coordinates of the mass into collecting matrix, then use this matrix do the PCA.

We use SVD method to perform PCA in each collecting matrix. The SVD will show us the most significant principle components and how those components' level of importance. That also can show us the reduction of dimension by PCA and our system after PCA still remain the pattern of the mass' motion, a simple harmonic motion. In test 3 and 4, we will see multiple level of noise, such as rotation, induced and how PCA deal with it.

Theoretical Background

First, PCA. Principal component analysis (PCA) is a statistical procedure that uses an orthogonal transformation to convert a set of observations of possibly correlated variables into a set of values of linearly uncorrelated variables called principal components. 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), and each succeeding component in turn has the highest variance possible under the constraint that it is orthogonal to the preceding components. The resulting vectors (each being a linear combination of the variables and containing n observations) are an uncorrelated orthogonal basis set. PCA is sensitive to the relative scaling of the original variables. (Wiki). We here use PCA to reduce the dimension of video frames and isolate the lower dimension true motion of the mass, which is one of the main principal components after PCA. However, PCA is sensitive to large variance and consider them as important. So, PCA may not always get the result we want.

SVD, The Singular Value Decomposition, is an operation finds the diagonal values of variance in order form largest to smallest.

$$A = U\sum V^*$$

Here, U and V are unitary matrix and $\sum$ is diagonal matrix.

Algorithm Development

On this homework, we need to get the coordinates of the mass in each frame first. We use **im2wb** with a sensitivity level to remove the color information. Also, we use an area section filter to section out the image area we don't want to analyze. So, we can focus on the mass movement rather than other objects.

```
%filter for section out other objects.
width = 50;
filter = zeros(480,640);
filter(300-2.6*width:1:300+2.6*width, 350-width:1:350+width) = 1;

X1 = vidFrames1_1(:,:,:,j);
figure(1)

%turn image into binary with gray scale filter.
level = 0.95;
X1b = im2bw(X1,level);
X1b = double(X1b);
X1b = X1b.*filter;
```

Then in order to find out the motion of the mass, we use **bwlabel** to label out the mass and **regionprops** to find centroid of each small section of the mass image data.

```
%label out section we need and find their centroid, store all objects in a
%stats array.
bw = bwlabel(X1b,4);
stats = regionprops(bw, 'BoundingBox', 'Centroid');
```

Then, we average their location to find a coordinate for this mass in this frame and store them into the position data matrix for this camera.

```
%initial center find and find center.
centerX = 0;
centerY = 0;
for object = 1:length(stats)
        bc = stats(object).Centroid;
        centerX = centerX+bc(1);
        centerY = centerY+bc(2);
end
hold off
centerX = centerX/length(stats);
centerY = centerY/length(stats);
%store center value into data array.
data1 = [data1;centerX,centerY];
```

We profom this operation for three cameras' video.

After collecting the position, we find out the frame number is different for three cameras. So we sync those three videos'data by align the first lowest point of the mass motion and crop them with length of the smallest video.

```
%clean and format datapoint

[M,I] = min(data1(1:25,2));
data1  = data1(I:end,:);
[M,I] = min(data2(1:25,2));
data2  = data2(I:end,:);
[M,I] = min(data3(1:25,1));
data3  = data3(I:end,:);
```

```
data2 = data2(1:length(data1), :);
data3vter = data3vter(1:length(data1), :);
```

Then we use SVD to perform PCA for all postion data.

```
dataAll = [data1';data2';data3vter'];
% Compute data size
[M,N]=size(dataAll);
% Compute mean for each row and subtract mean with all the data.
mean=mean(dataAll,2);
dataAll=dataAll-repmat(mean,1,N);
% Deploy the SVD
[u,s,v]=svd(dataAll'/sqrt(N-1));
%diagonal variances
lam=diag(s).^2;
%the principal components projection
Y= dataAll' * v;
%find signal
sig=diag(s);
```

Repet for four different test. The section filter and gray scale filter will have minor adajst for for differetnt test.

Computational Results

Test 1(ideal case):

We can see that PAC did a good job and the first principal component is the motion of the mass. Results in MATLAB CODE part.

Test 2(noisy case):

We can see the raw data indeed have many noises. After PCA, we still can see a pattern of the oscillation of the mass. Results in MATLAB CODE part.

Test 3(horizontal displacement):

We see more principal components have a large level of variance. Which cause our motion of the mass After PCA is not ideal. Results in MATLAB CODE part.

Test 3(horizontal displacement + rotation):

We see more principal components have a large level of variance. Which cause our motion of the mass After PCA is not ideal. Results in MATLAB CODE part.

Summary and Conclusions

We observed the power of PCA and find the limit of it. The PCA is convenience for simple variance in linear analysis. However, the larger variance induced, the less accurate of PCA is.

MATLAB CODE: