Principal Component Analysis with SVD

Rachel Carroll[*]

03/17/2020

Abstract

In this paper, principal component analysis (PCA) is used to extract physical properties of a harmonic oscilator using image data. The system set up was a paint can attached to a spring. Videos of simple oscilations, pendulum motion, and rotation of the system were captured. Using color and brightness patterns of the video pixels, the paint can's location was tracked frame by frame. Using resulting x and y coordinate vectors, the SVD was used to perform PCA and identify a coordinate system that best represents the physical motion of the paint can. This method successfully identified the expected movement basec on known physical properties. The accuracy of the results depended on the amount of noise in the data (mainly in the form of camera shake) and the quality of the tracking algorithm.

1 Introduction and Overview

This paper uses principal component analysis (PCA) to measure the physical behavior of a harmonic oscillator system. The system used comprised of a paint can attached to a spring. Video recordings of the following four cases were analyzed

1. **Simple harmonic motion (ideal case):** A small displacement of the mass in the z direction.
2. **Simple harmonic hotion (noisy case):** The same as above but with camera shake in video recording.
3. **Horizontal displacement:** In this case, the mass is released off-center so as to produce motion in the x—y plane as well as the z direction. Thus there is both a pendulum motion and a simple harmonic oscillations.
4. **Horizontal displacement and rotation:** In this case, the mass is released off-center and rotates so as to produce motion in the x—y plane, rotation as well as the z direction. Thus there is both a pendulum motion and a simple harmonic oscillations.

For each case, three videos were taken from different angles. In each video, the motion of the paint can was tracked, frame by frame, using color and brightness patterns of the pixels. These measurments were stored in x and y coordinate vectors, representing the motion over time. For each case there were a total of six vectors (an x and y vector for each of the three videos).

The properties of the SVD were leveraged to remove redunences in the patterns of motion captured in the the three videos. By doing so, our initially oversampled data set was reduced to highlight the primary components of the paint can's motoin. By taking the SVD we were able to see which direction were most relevant and how system moved in that direction over time.

2 Theoretical Background

Introducing the Singular Value Decomposition (SVD):
The SVD is a very valuable tool in computational data analysis because it extracts relevant correlations and

[*]https://github.com/rachel-carroll/

variances across a data set, and in turn, highlights its significant behaviors. Given a matrix A, the SVD is defned as follows

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

where U and V are othonormal and Σ is diagonal. To understand this representation, we will first consider how a matrix A behaves as a linear operator. Consider the standard linear equation $Ax = b$. This operation can be thought of as a projection of x onto the range of A. Physically, this occurs by rotating and stretching x, transformation into the vector b.

Consider the set of vectors $\{v_1, v_2, \dots, v_n\}$ that form an orthonormal basis. When any of these vectors are multiplied by A, they will be rotated and stretched. Letting V be the unitary matrix with columns $v_1, v_2, \dots, v_n$ we can represent this behavior with

$$AV = U\Sigma$$

Where U is an orthonormal matrix which rotates a vector but does not change its size. On the other hand Σ is a diagonal matrix so it stretches a vecor but does not rotate it. Since V is unitary we can rewite the equation in the SVD form as shown above.

Now the equation $Ax = b$ can be rewritten as $U\Sigma V^* x = b$. This breaks down the fundamental behavior as A acts on x by rotating, stretching, and rotating again. The columns of the U and V matrix are called the **singular vectors** of A and the diagonal elements of Σ are the **singular values** of A.

An important feature of the SVD is that **any** matrix A can represented this way. The key to this is the use of two bases, U and V. For example, an eignevalue decomposition $A = S\Lambda S^{-1}$, can only work when A is a normal matrix since we are restricted to only the basis of eigenvectors.

The Covariance Matrix:

For any given matrix $A^{m\times n}$, its covariance marix, C_A is given by

$$C_A = \frac{1}{n-1} A^T A$$

The diagonal terms of C_A are the variances of the columns of A. Large values represent significant dynamics. The off-diagonal terms are the covariances between the columns. Large off diagonal entries represent a high-degree of redundancy, whereas a small off-diagonal coefficients means there is little redundancy in the data, i.e. they are statistically independent. Notice

$$\begin{aligned} A^T A &= (U\Sigma V^*)^T (U\Sigma V^*) = V\Sigma^T U^* U\Sigma V^* = V\Sigma^2 V^* \\ &\implies A^T A V = V\Sigma^2 \end{aligned}$$

and

$$\begin{aligned} AA^T &= (U\Sigma V^*)(U\Sigma V^*)^T = U\Sigma V^* V\Sigma U^* = U\Sigma^2 U^* \\ &\implies A^T A U = U\Sigma^2 \end{aligned}$$

This shows that the columns of V are the eigenvectors of $A^T A$ and the columns of U are the eigenvectors of AA^T and the eigenvalues of the covariance matrix equal the squares of the singular values.

This result is important because it shows how the SVD can diagonalize the covariance matrix which allows us to see where the data has high or low redundency. So the relatively largest singular values will indicate significant dynamics and the associated eigenvectors will map what those dynamics are.

Principal Component Analysis

Principal component analysis is a method that uses the realtionship of the SVD and the covariance matrix into practice. We can use the SVD to extract information of a low dimensional system from a oversampled dataset. In this experiment the dataset is a $6 \times m$ matrix where the columns are the x and y vectors from the three videos and m is the number of frames. Taking the SVD of the the system generates the following matrices:

- U – An $m \times 6$ orthonomal matrix. Each column represents the paint cans movement in time.
- Σ – A 6×6 diagonal matrix with elements whose relative magnitudes highlight significant dynamics.
- V – A 6×6 unitary matrix whose columns provide the x and y direction of motion.

The relative magnitude of the singular values, $\sigma_i \in \Sigma$, will tell us how many significant modes exist in our system (in this case, the modes represent dimensions of motion). The larger the value, the more significant. Then, we can see what the behavior of the significant modes look like by looking at the cooresponding columns of U and V. The V vectors indicate the direction of the motion and the U vectors identify the movement over time in that direction.

3 Algorithm Implementation and Development

The following steps were performed in MATLAB

1. Track x and y coordinates of the point can in each frame of each video (refered to as video a, b, and c)
2. Subtract the mean from each vector. This creates an "equal playing field" so the motion of each vector is compared without bias based on starting loaction realtive to the different camera angles.
3. Store the x and y vectors of each coordinate system in a matrix of the form

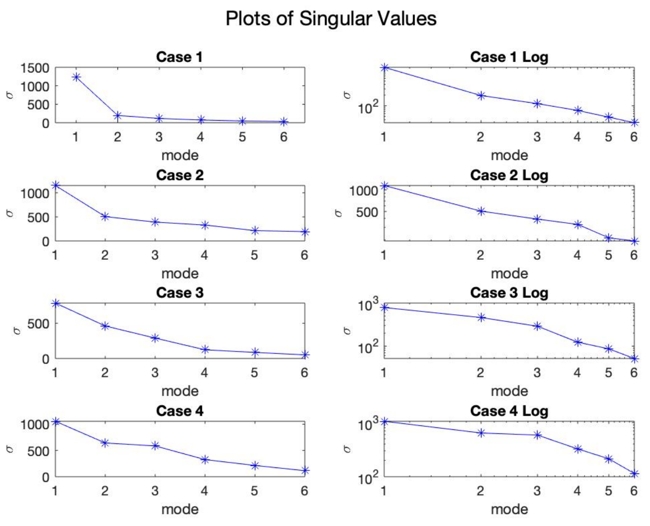
$$A = \begin{bmatrix} \\ x_a & y_a & \cdots & x_c & y_c \\ \\ \end{bmatrix}$$

   Where x_i, y_i are the x and y coordinate vectors from video i ($i = a, b, c$). Each vector contains the number of

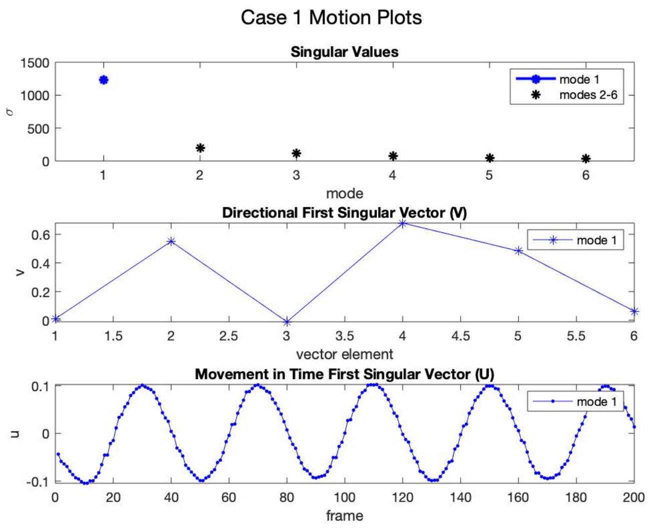
4. Caculate the SVD.
5. Identify the significant modes by finding the relatively highest singular values
6. Plot the V columns of the significant modes. This represents the direction of that mode.
7. Plot the U columns of the significant modes. This represent the motion in time of the mode (in the direction if the corresponding V column).

4 Computational Results

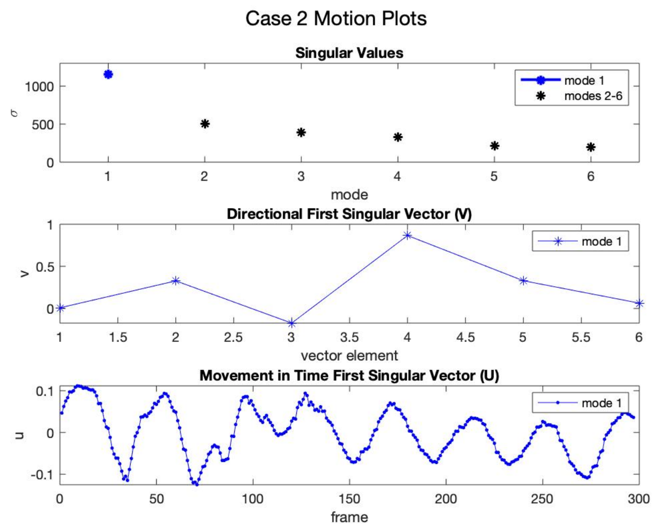
Comparison of singular values across cases: The plot below shows plots of the singular values for all four cases. We will go through case by case



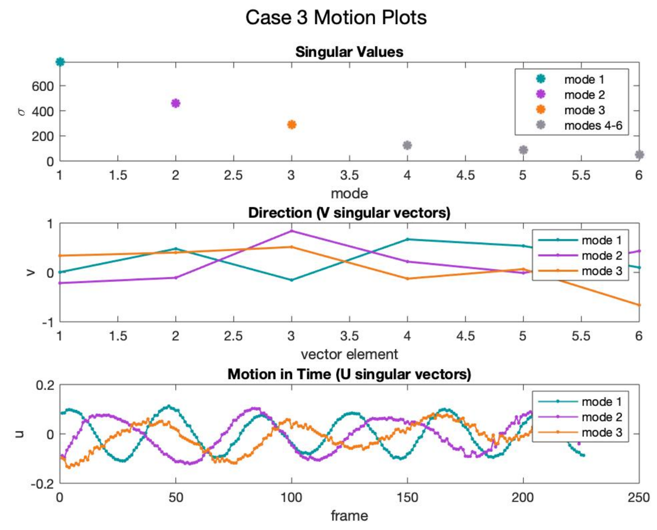
Case 1: This case is the ideal situation indeed. With little noise “in the form of camera shake” it is quite clear that there is only one significant mode. This is exactly what we expect to see as this case represents a simple harmonic oscillator which should only be moving in one direction (up and down).



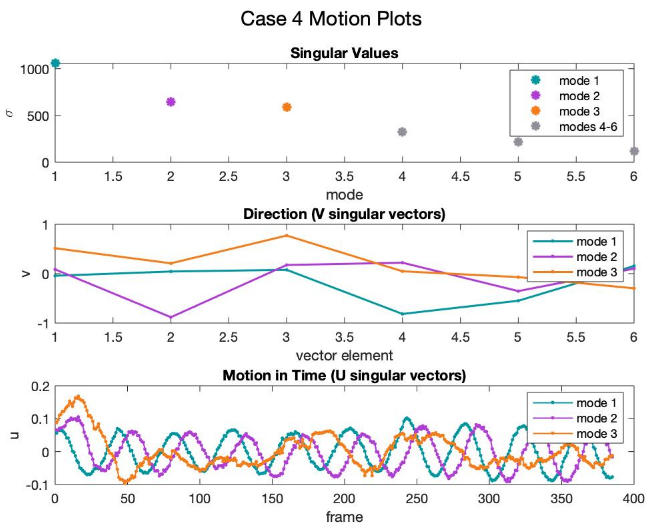
Case 2: This result is also what we expect to see. The motion of the system should be the same as case 1 and only have one significant mode. However, with the introduction of additional noise through the camera shake, the singular values for modes two through six increase. In this case the noise is not too drastic so we can still see that mode one stands out. Having said that, it is apprent that with more noise, we could not identify the primary coordinates with confidence.



Case 3: The significant modes here are less obvious but it looks like there is some significance in the first three. Here, the loglog plot helps demonstrate more of a drop off after node three (see singular value plot). Notice, that in this case we were expecting only two significant modes (up and down and side to side). However, there is inherent human error in deploying the system, holding the camera, as well as challenges in traking the paint can perfectly. So it is not too surprising that three dimensions of motion were highlighted.



Case 4: Here the expectation is three significant coordinated of motion (modes). There is noise that brings up the later modes, but the first three stand out. Also, compared to case three, mode three appears to be more significant in this case which we expect to see. Again, comparing the log plot of case three and four demonstrate this well.



A final note to add about all of thes case figures, is that the third plot depicts harmonic motion (some more clearly than other depending on the data quality). This is what we expect to see in each direction given that the experiment used rotations, pendulums, and oscilations which all create harmonic motion.

5 Summary and Conclusions

This experiment boils down to a fundamental concept in computational data science. That is, identifying an appropriate coordinate system in which to analyze data. For each case, we used an overrepresented data set from three, two-dimensional coordinate systems (i.e. the videos). From this we extracted the "true" three dimensional movement of the paint can. Because we live in a three dimensional world with known physical properties, this seems obvious. However, in practice, the "true" coordinate system is not always so apparent. This demonstrates the SVDs power in identifying such coordinate systems and, in turn, finding the significant behavior otherwise hidden in an over or under represented data set.

6 Appendix A MATLAB functions used and brief implementation explanation

1. track_pt - Custom function that searches for the brightest white point in a specified window frame by frame. The output is the x and y coordinate vectors tracking the paint cane throughout the video.
2. track_red_pt - Like track but searched for red points
3. track_w_red_pt - A combination of track_pt and track_red_pt. It searches for the brightest white above a given threshold. If that threshold is not reached, it searches for the reddest point.
4. show1stframe - Custom function that dislays the first frame of the video with an initial tracking point.
5. imshow - MATLAB function that shows the image of a given frame of the video
6. track_vid - Custom function that plays the video along with a tracking point highlighting the x and y coordinates from in the track_ functions. This was used to watch every video with the tracking results to make sure the point remained on the paint can, follong its path of motion.

7 Appendix B MATLAB codes

See https://github.com/rachel-carroll/
Repository: AMATH582
Folder name: Principal_Component_Analysis_Report
File name: PCA_code.m (Published version “PCA_code.pdf)