Coding Project 3: Principal Component Analysis of a Mass-on-a-spring System

Tyler Shakibai

Abstract

We examine the dynamics of a mass in motion on a spring with four separate tests being recorded from different three angles, each providing a different perspective of the object and accompanying level of noise. Each angle records measurements in its own coordinate system relative to the camera, making it difficult to accurately characterize the motion of the object. The varying levels of disturbance the camera is subjected to are stable, shaking, displaced, and rotating. We employ singular value decomposition to project the information from the tests into lower-dimensional representations. Our goal is to extract the principal components of the object's motion in order to create lower-rank approximations that still comprise most of the energy, or information, of the system.

1 Introduction

Principal component analysis (PCA) is a technique in dimensionality reduction for analyzing datasets of massive dimensionality and size. Dimensionality is defined as the number of characteristic features in some mathematics space. For example, a vector space that is in $\mathbb{R}^3$ has a dimensionality of three. We can also interpret the size of a dataset to be the number of rows, and the dimension to be the number of columns. PCA is used to increase the interpretability of the data it is applied to by discarding redundant or unnecessary features. PCA maintains as much information as possible by only preserving the dimensions that are crucial to the qualitative behavior. PCA is most applicable when the data in question is of high dimension, i.e.

consists of many variables, and is most effective when multi-colinearity exists between the variables.

A mass-on-spring system is a dynamical system that models the oscillatory behavior of the motion of an object connected to a spring with tension. The oscillations are a product of kinetic energy being converted to potential energy in the spring and vice versa. The second-order ordinary differential equation that governs this system can be written as

$$my'' + by' + ky = 0$$

where m is the mass, k is the spring constant, and b is the damping constant. The solution to this ODE takes the form

$$y = A \sin \gamma t + B \cos \gamma t$$

where constants A, B, and γ are determined by constants m, k, and b.

In this report, we will discuss how singular value decomposition (SVD) and principal component analysis can be utilized to lend insight into the underlying dynamics of a system we have measurements of. We will then apply these techniques to produce low-rank approximations that still retain high fidelity when modeling the motion of our mass in four different recording scenarios: ideal, noisy, displaced, and rotating. The goal of applying these methods is to identify and minimize redundancy among the features of our measurements.

2 Theoretical Background

In this section, we will delve into the theoretical underpinnings of the aforementioned dimensionality reduction techniques. This discussion will be focused primarily on what the decompositions are doing at the level of the matrices being manipulated and how this affects the information stored in them.

2.1 Singular Value Decomposition

Singular value decomposition is a matrix decomposition method that is in and of itself a form of principal component analysis. This decomposition can notably be represented as two matrix transformations, a rotation and a scaling effect (either stretch or compress).

The SVD, which exists for any matrix $A \in \mathbb{C}^{m \times n}$ is defined as

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

where U is a unitary matrix describing each principal component in time, Σ is a diagonal matrix whose elements are a sorted vector of singular values, which represents the amount of energy or significance of each component, and V^* is the conjugate transpose of V which is the unitary matrix which contains the unit vectors for the principal components of the new basis, also known as the orthonormal basis. These components of the SVD have desirable properties such as orthogonality and diagonality which make them easy to work with.

2.2 Rank-n Approximation

A rank-n approximation is a low-rank approximation of dimensionality n that can be constructed using the singular value decomposition. SVD orders the singular values in the diagonal matrix Σ from greatest to least. Since these singular values indicate the amount of energy, or how much information, is stored in the corresponding component of these singular values, we always know how much energy is stored in a rank-n approximation for $n \in \mathbb{N}$. The amount of energy stored in any rank-n approximation is the sum of the nth singular values.

Since the vector of singular values in Σ is strictly decreasing, we know that each additional rank of our approximation will provide diminishing returns to the total amount of energy it contributes to the reconstructed matrix. For example, the difference in energy between a rank-2 and rank-1 approximation is greater than or equal to the difference between and rank-3 and rank-2 approximation. The most prevalent modern-day application of rank-n approximations is lossy compression, where matrices of data are approximated with a lower-rank matrix that requires fewer values to be stored than the original.

2.3 Principal Component Analysis

Principal component analysis is a dimensionality reduction technique used to make convoluted data more interpretable. The goal of applying PCA is to find a low-dimensional representation of some higher-dimensional but feature-redundant information. We can define the redundancy of a dataset as observations that overlap in that they contribute to variance along the same axis. PCA is also capable of reducing noise that would other obscure meaningful information from the measurements of our mass-on-spring system.

An important notion in understanding why principal component analysis is useful is covariance, which measures the probabilistic independence or dependence between two variables. A covariance matrix is a matrix that can be applied to a set of data in matrix form to determine which observations within the dataset can be omitted because they are redundant. In other words, it defines how much certain observations correlate with other observations in the same set of data. The covariance matrix can be defined as

$$C_X = \frac{1}{n-1} X X^T$$

where X represents the data of interest. The covariance matrix C_X is square and symmetric, and its diagonal elements correspond to the variance of each observation.

3 Numerical Methods

Since the numerical process is the same for each method, know that to perform this process for each test scenario, one must simply repeat the following method. First, we need to load and prime our data to ensure the sequences from the three cameras are of the same length. Next, we subtract the mean from each row vector to normalize the data. We now take the singular value decomposition to give decomposed matrices S, Σ, and V^*. We normalize our singular values by diving the diagonal vector of Σ by the L2 norm of Σ.

Now that we have successfully computed the SVD, that analysis part begins. Since we would like to know what are appropriate values of n for later rank-n approximations, we plot the logarithmically scaled energies which correspond to the singular values in Σ. To determine how many singular values we need to maintain reasonably high fidelity in our approximation, we look to see how many singular values are required to incorporate 85% of the energy of the original matrix. Once we have computed the optimal value for n, we are simply matrix multiply S, Σ, and V^*, but with only n columns of S, n diagonal entries of Σ, and n columsn of V^*.

4 Results

Our results suggest that we were able to accurately capture the sinusoidal dynamics of our mass-on-spring system. Using figures 1, 3, 5, and 7, we were able to inform what best value of n to use for our rank-n approximations in figures 2, 4, 6, and 8. In each test, the approximation was able to model a sinusoid which matches our intuition of the oscillatory behavior of a spring attached to a weight. Furthermore, each curve is appropriately altered to align with the test, for example, the rank-n approximation for test two appears to have more noise than the others.

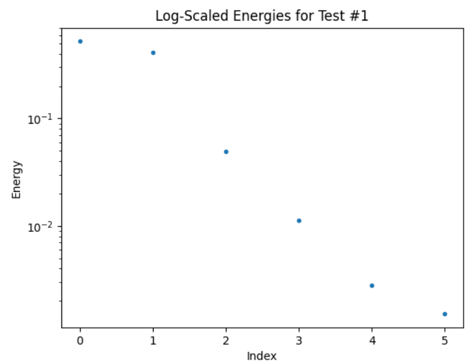

Figure 1: Plot of the logarithmically scaled energies corresponding to the singular values of test #1.

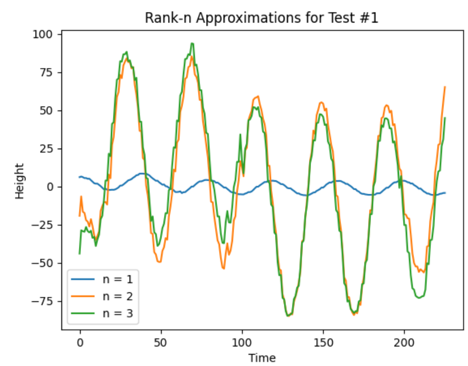

Figure 2: Plot of the rank-n approximations for $n \in [1, 3]$ for test #1.

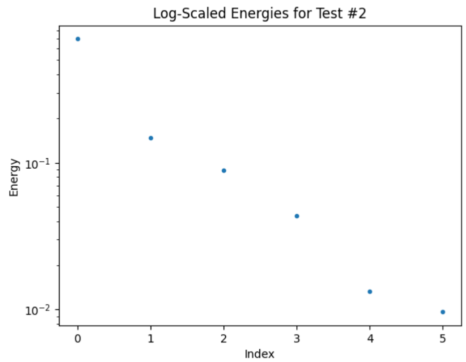

Figure 3: Plot of the logarithmically scaled energies corresponding to the singular values of test #2.

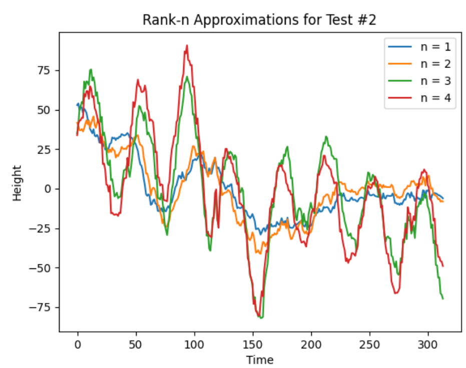

Figure 4: Plot of the rank-n approximations for $n \in [1, 4]$ for test #2.

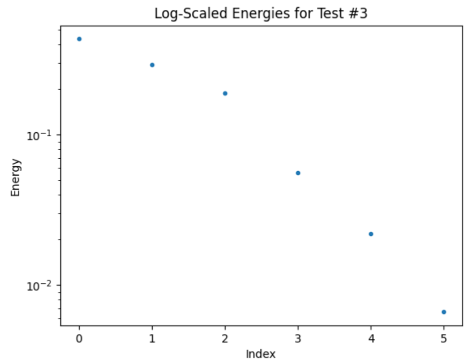

Figure 5: Plot of the logarithmically scaled energies corresponding to the singular values of test #3.

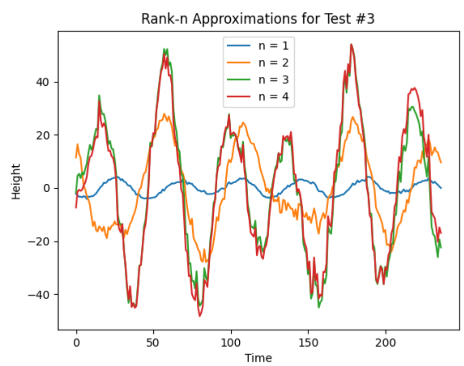

Figure 6: Plot of the rank-n approximations for $n \in [1, 4]$ for test #3.

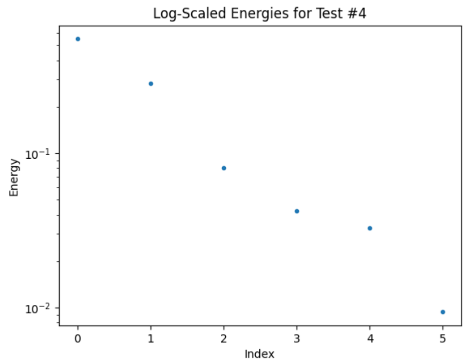

Figure 7: Plot of the logarithmically scaled energies corresponding to the singular values of test #4.

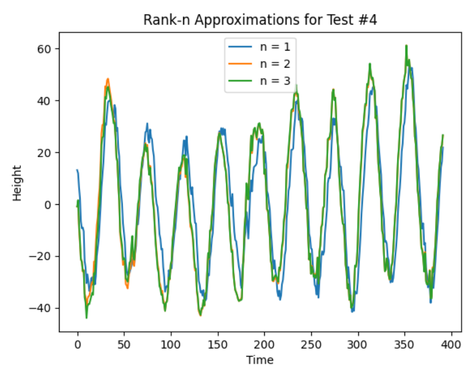

Figure 8: Plot of the rank-n approximations for $n \in [1, 3]$ for test #4.

Surprisingly, our use of principal component analysis was extremely effective in test four despite the rotation applied to the recording of our measurements. In each even-numbered figure, we see our approximation become increasingly accurate as more of the energy of the system is made available via matrix multiplying by the nth principal component.

5 Conclusion

Overall, our numerical scheme for performing dimensionality reduction on our measurements of the mass-on-spring system was successful. We were able to accurately extract accurate positional data despite the varying noise from the four tests and the results were satisfactory in each scenario. Principal component analysis was employed to extract a low-dimensional representation of our system. We used singular value decomposition to decompose our concatenated matrices of the varying camera angles and extract the principal components in order to reduce our feature space. The choice of n for the rank-n approximation matched expectation in that we selected the fewest singular values to capture the oscillatory behavior of the system.

Acknowledgment

I attended office hours with the course's TAs, Katie and Michael, and heavily utilized the Numpy and Scipy Documentation.