

# AMATH 482/582: HOME WORK 2

SAMIP KARKI

*Applied Mathematics, University of Washington, Seattle, WA*  
*karki1@uw.edu*

ABSTRACT. The singular value decomposition is used in many data applications. In this work, I use singular value decomposition and principle node analysis for dimension reduction. Taking position data from a robot performing different tasks such as walking, running, and jumping, I find that the position measurements, which have dimension of 114, can be accurately approximated with as little as 10 principal component modes. I make a classifier which assigned new measurements labels (walking, running, or jumping) by comparing the distance in the principal component mode basis between the new measurement and the centroids of each of the labels.

## 1. INTRODUCTION AND OVERVIEW

In this project, I have data of measurements representing a robot performing three movements: walking, jumping, and running (figure 1). The raw data comes in the form of  $(x, y, z)$  coordinates of various joint at various time-steps. I analyze the data using PC mode analysis and create a classifier which can assign the the labels of walking, jumping, running to each time-step.

## 2. THEORETICAL BACKGROUND

When analyzing data with high dimension, it is often useful easier to consider lower dimensional representation of the data. Consider a sequence of measurements represented by the columns of the  $n \times m$  matrix  $A$ . The singular value decomposition of  $A$  is given by:

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

---

*Date:* February 7, 2024.

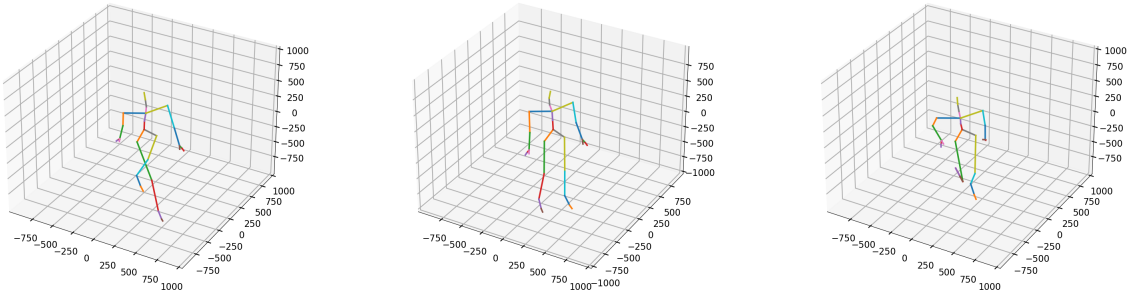


FIGURE 1. Examples of the each of the three labels, from left to right: walking, jumping, and running.

### Cumulative Sum of Energy of Principal Components

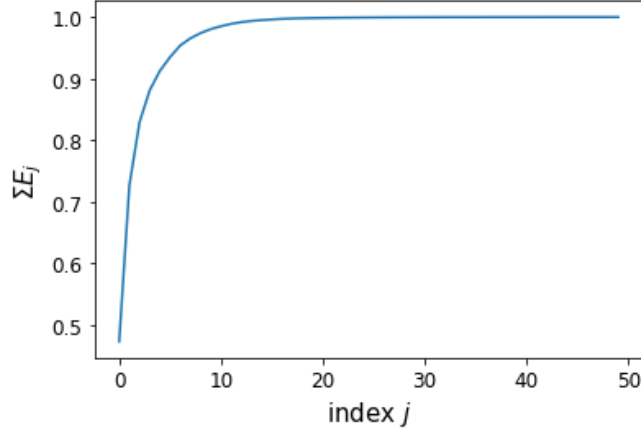


FIGURE 2

notice, the  $i$ -th column of  $A$  is given by:

$$a_i = \sum_{i=0}^m c_i u_i$$

where  $u_i$  are the  $i$ -th column of  $U$  and the coefficients  $\vec{c} = [c_0, c_1, \dots, c_m]^T$  given by the  $i$ -th column of  $\Sigma V^T$ . It can be seen that  $a_i$  can be expressed in the basis of  $u_1, u_2, \dots, u_m$ , also called the principle component (PC) modes. By truncating this sum, we obtain a lower dimensional approximation for  $a_i$ .

The columns of  $U$  are ordered in such a way that the modes which contribute the most to  $a_i$  appear first in the sum. Therefore, we can capture most of the behavior of  $a_i$  with a relatively small number of PC modes. Instead of doing analysis on  $A$ , we can instead analyze the first  $k$  rows of  $\Sigma V^T$ , which represents the projection of  $A$  onto the first  $k$  PC modes.

### 3. ALGORITHM IMPLEMENTATION AND DEVELOPMENT

All the computation and plots made in this project is done with the canonical python packages: `numpy`, `matplotlib`. SVD was computed using `numpy.linalg.svd`.

### 4. COMPUTATIONAL RESULTS

Our training data consists of 500 time steps of the humanoid robot performing each movement: walking, jumping, and running. There are Let  $\tilde{X}_{train}$  be the  $114 \times 1500$  matrix where the rows represent the  $(x, y, z)$  coordinates of the 38 joints of the robot the columns represent each time-step.  $X_{train}$  is the matrix where each row of  $\tilde{X}_{train}$  is subtracted by the mean of that row. I take the singular value decomposition of  $X_{train}$

$$X_{train} = U \Sigma V^T$$

to obtain the basis of principle component (PC) modes (columns of  $U$ ) and the coefficients ( $\Sigma V^T$ ).

The energy,  $E_k$  of the  $k$ -th PC mode is given by

$$E_k = \frac{\sigma_k^2}{\sum_i \sigma_i^2}.$$

The cumulative sum of the Energy of the first  $j$  PC modes of  $X_{train}$  are graphed in figure 2. The first 1, 2, 4, and 6 PC nodes are required to accumulate 70, 80, 90, and 95 percent of the total

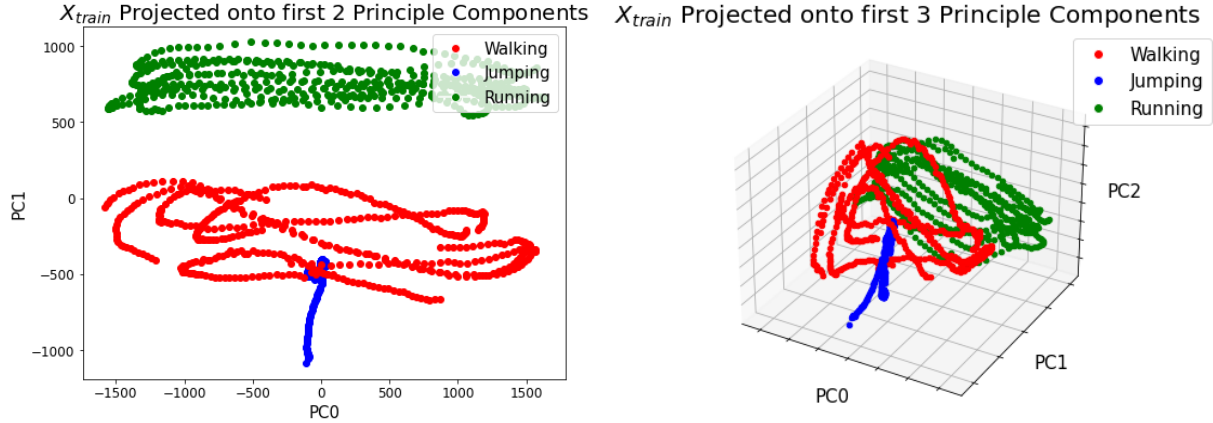


FIGURE 3. Projection of the columns of  $X_{train}$  on to the first 2 (left) and first 3 (right) principal component modes of  $X_{train}$ . The colors represent each of the three label, walking, jumping, running.

energy, respectively. It can be concluded that as low as 6 PC is all that is needed to sufficiently capture the characteristics of the measurements in  $X_{train}$ .

The projection of the columns of  $X_{train}$  onto the first 3 PC modes are given in figure 3. It can be seen that each label, walking, jumping, and running, has a discernible path which is taken in the plane of the first 2 and 3 PC modes.

Next, I leverage PC analysis to create a simple classifier. My goal is to take new measurements  $\tilde{X}_{test}$  and to correctly assign labels to them  $Y_{guess}$ . The centroid of each label in  $X_{train}$  is found,  $C_w$ ,  $C_j$ , and  $C_r$  for walking, jumping, and running respectively. The algorithm for the classifier is as follows:

- Transform  $\tilde{X}_{test}$  by subtracting the row mean from each row and change to PC basis by applying  $\hat{U}^T$ , where  $\hat{U}$  is truncated to only have the first 114 columns of  $U$ . This new matrix will be called  $X_{test,PC}$
- For each column in  $X_{test,PC}$  called  $x_i$ , find the distance between  $x_i$  and each of the centroids  $C_w$ ,  $C_j$ , and  $C_r$  using the first  $k$  PC mode coefficients of  $x_i$  and the centroids.
- Assign the label in  $Y_{guess}$  for which centroid  $x_i$  was closest to.
- Do this for various values of  $k$ .

On the left of figure 4, I plot how accurate the classifier is at labeling the training data  $\tilde{X}_{train}$  as the number of PC modes included  $k$  changes. On the right of figure 4, I use the classifier to label a new data set of measurements  $\tilde{X}_{test}$  which includes 100 measurements of each of the labels. We can see that the performance of the classifier does not get any better with more than 10 PC modes included. This is consistent with figure 2, where most of the energy of  $X_{train}$  was within the first 10 PC modes.

## 5. SUMMARY AND CONCLUSIONS

In this project, I explored how principle component analysis can be used to reduce the dimension of data. I saw that 10 principal component modes was sufficient to capture most of our robot position data. I was able to use PC analysis to create a classifier and tested my classifier on new data. I took AMATH 584 Numerical Linear Algebra, and it feels like I am applying much of the theory I learned from that courses into a hands-on project in this course.

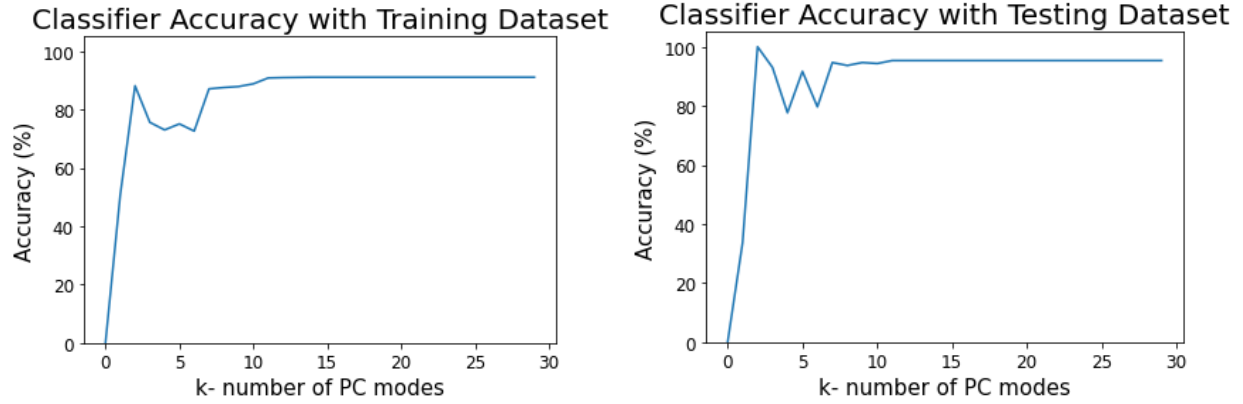


FIGURE 4. The performance of centroid distance classifier for various number of PC modes included,  $k$ . On the left, the classifier's performance in labeling  $X_{train}$ . On the right, the classifier's performance in labeling  $X_{test}$ .

#### ACKNOWLEDGEMENTS

I'd like to thank Prof. Shlizerman who was available to answer my questions during office hours.